Advanced NetFlow Deployment

BRKNMS-3006

Benoit Claise

Cisco Networkers 2007

HOUSEKEEPING

- We value your feedback, don't forget to complete your online session evaluations after each session and complete the Overall Conference Evaluation which will be available online from Friday.
- Visit the World of Solutions on Level -01!
- Please remember this is a 'No Smoking' venue!
- Please switch off your mobile phones!
- Please remember to wear your badge at all times including the Party!
- Do you have a question? Feel free to ask them during the Q&A section or write your question on the Question form given to you and hand it to the Room Monitor when you see them holding up the Q&A sign.

Session Abstract

- This advanced session focuses on the latest NetFlow developments: new features, the latest studies about sampling, the NetFlow version 9 and its standardization at the IETF. Specifically, the new Flexible NetFlow feature will be covered in detail. The technical details of the new features will be addressed with configuration examples, show commands, tricks, and best practice advices. Scenarios such as NetFlow for security and NetFlow for capacity planning are specifically covered. A few implementation details of the different Cisco platforms will be provided, with a little bit of troubleshooting.
- This session is designed to be particularly useful for attendees working in the following areas: enterprise, service provider and NREN experts, engaged in designing, maintaining, and troubleshooting security, capacity planning, and accounting solutions. Attendees should be familiar with network management basics, and should already have some understanding of NetFlow, perhaps by already having taken the introductory session

This Tutorial Is …

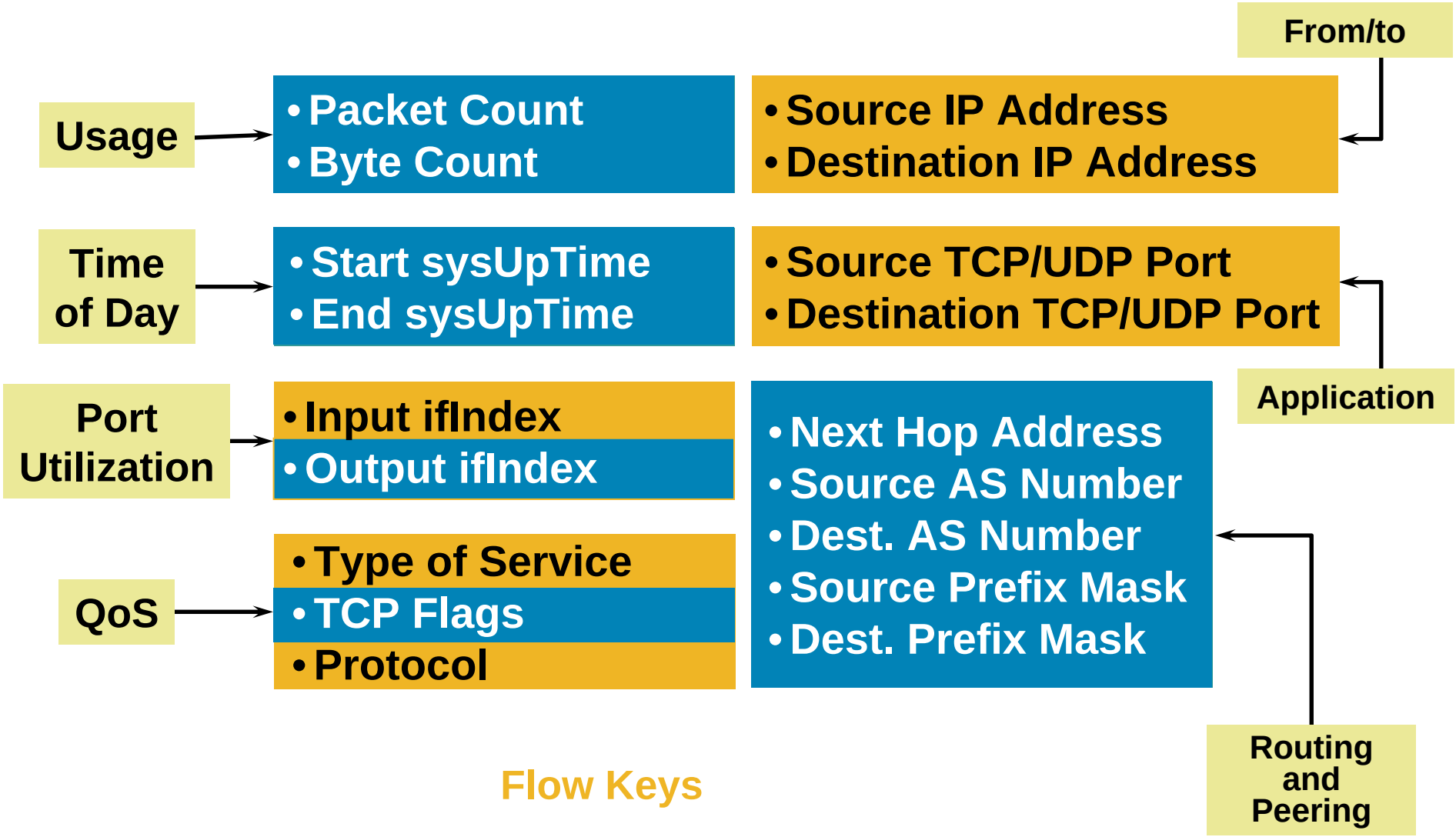
- Not about
  - A level 1 type of presentation
    - Networkers On Line session “Introduction to IP Accounting and Netflow” (NMS-1532)
  - Marketing slides
  - The NetFlow collector details
  - The ecosystem partners applications and mediations
  - Many platform specific details
- About
  - New features, mainly in the software platforms
  - Advanced information
  - And scenario…
    - Assuming the NetFlow basics are known

Agenda

- Introduction
- NetFlow Version 9
- New Features
- Flexible NetFlow
- NetFlow for Security
- NetFlow for Capacity Planning
- Platforms Specific
- NetFlow Ongoing Developments

Introduction

Version 5 Flow Format



NetFlow Cache Example

1. Create and update flows in NetFlow cache

SrcIf	SrcIPadd	DstIf	DstIPadd	Protocol	TOS	Flgs	Pkts	Src Port	Src Msk	Src AS	Dst Port	Dst Msk	Dst AS	NextHop	Bytes/ Pkt	Active	Idle
Fa1/0	173.100.21.2	Fa0/0	10.0.227.12	11	80	10	11000	00A 2	/24	5	00A2	/24	15	10.0.23.2	1528	1745	4
Fa1/0	173.100.3.2	Fa0/0	10.0.227.12	6	40	0	2491	15	/26	196	15	/24	15	10.0.23.2	740	41.5	1
Fa1/0	173.100.20.2	Fa0/0	10.0.227.12	11	80	10	10000	00A 1	/24	180	00A1	/24	15	10.0.23.2	1428	1145.5	3
Fa1/0	173.100.6.2	Fa0/0	10.0.227.12	6	40	0	2210	19	/30	180	19	/24	15	10.0.23.2	1040	24.5	14

2. Expiration

- **Inactive Timer Expired (15 Sec Is Default)**
- **Active Timer Expired (30 Min Is Default)**
- **NetFlow Cache Is Full (Oldest Flows Are Expired)**
- **RST or FIN TCP Flag**

SrcIf	SrcIPadd	DstIf	DstIPadd	Protocol	TOS	Flgs	Pkts	Src Port	Src Msk	Src AS	Dst Port	Dst Msk	Dst AS	NextHop	Bytes/ Pkt	Active	Idle
Fa1/0	173.100.21.2	Fa0/0	10.0.227.12	11	80	10	11000	00A2	/24	5	00A2	/24	15	10.0.23.2	1528	1800	4

3. Aggregation

No

Yes

4. Export version

Non-aggregated flows—export version 5 or 9

E.g. Protocol-Port Aggregation Scheme Becomes

Protocol	Pkts	SrcPort	DstPort	Bytes/Pkt
11	11000	00A2	00A2	1528

Aggregated Flows—Export Version 8 or 9

5. Transport protocol

Export Packet

‘show ip cache flow’

Packet Sizes

# of Active Flows

Rates and Duration

Flow Details

```
router# show ip cache flow
IP packet size distribution (85435 total packets):
   1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
   .000 .000 .125 .125 .250 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

    512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
   .000 .000 .000 .000 .500 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 278544 bytes
  2728 active, 4368 inactive, 85310 added
  463824 ager polls, 0 flow alloc failures
  Active flows timeout in 30 minutes
  Inactive flows timeout in 15 seconds
  last clearing of statistics never
Protocol         Total     Flows   Packets Bytes  Packets Active(Sec) Idle(Sec)
--------         Flows      /Sec     /Flow  /Pkt     /Sec     /Flow     /Flow
TCP-X                2       0.0         1  1440       0.0       0.0       9.5
TCP-other        82580      11.2         1  1440      11.2       0.0      12.0
Total:           82582      11.2         1  1440      11.2       0.0      12.0

SrcIf         SrcIPaddress    DstIf         DstIPaddress    Pr SrcP DstP  Pkts
Et0/0         132.122.25.60   Se0/0         192.168.1.1     06 9AEE 0007     1
Et0/0         139.57.220.28   Se0/0         192.168.1.1     06 708D 0007     1
Et0/0         165.172.153.65  Se0/0         192.168.1.1     06 CB46 0007     1
```

‘show ip cache verbose flow’

```
router# show ip cache verbose flow
IP packet size distribution (23597 total packets):
   1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
   .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

    512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
   .000 .000 .000 .000 1.00 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 278544 bytes
  1323 active, 2773 inactive, 23533 added
  151644 ager polls, 0 flow alloc
  Active flows timeout in 30 minut
  Inactive flows timeout in 15 sec
  last clearing of statistics neve
Protocol         Total     Flows   Packets Bytes  Packets Active(Sec) Idle(Sec)
--------         Flows      /Sec     /Flow  /Pkt     /Sec     /Flow     /Flow
                                      3.1      1  1440      3.1       0.0      12.9
Total:                                3.1      1  1440      3.1       0.0      12.9

SrcIf         SrcIPaddress    DstIf         DstIPaddress    Pr TOS Flgs Pkts
Port Msk AS                   Port Msk AS   NextHop            B/Pk  Active
Et0/0         216.120.112.114 Se0/0         192.168.1.1     06 00  10       1
5FA7 /0  0                    0007 /0  0    0.0.0.0              1440     0.0
Et0/0         175.182.253.65  Se0/0         192.168.1.1     06 00  10       1
```

Flow Rate and Duration

ToS Byte and TCP Flags

Destination Information

Source Mask and AS

NetFlow Export Version 5 and Main Cache Configuration Example

```
Router(config)# interface <slot/port/subinterface>
Router(config-if)# ip flow ingress
Router(config-if)# ip flow egress

Router(config)# ip flow-cache entries <number>
Router(config)# ip flow-cache timeout active <minutes>
Router(config)# ip flow-cache timeout inactive <seconds>

Router(config)# ip flow-export version 5 peer-as
Router(config)# ip flow-export destination 10.10.10.10 1234
Router(config)# ip flow-export source loopback 0
```

NetFlow Export Version 5 and Main Cache Configuration Example

```
Router # show ip flow export
Flow export v5 is enabled for main cache
  Exporting flows to 10.48.71.129 (9991)
  Exporting using source interface Loopback0
  Version 5 flow records
  1303552 flows exported in 332208 udp datagrams
  0 flows failed due to lack of export packet
  2 export packets were sent up to process level
  0 export packets were dropped due to no fib
  0 export packets were dropped due to adjacency issues
  0 export packets were dropped due to fragmentation failures
  0 export packets were dropped due to encapsulation fixup failures
  0 export packets were dropped enqueuing for the RP
  0 export packets were dropped due to IPC rate limiting
  0 export packets were dropped due to output drops
```

NetFlow Export Version 8 and Aggregation Configuration Example

```
Router(config)# ip flow-aggregation cache <cache type>
Router(config-flow-cache)# cache entries <number>
Router(config-flow-cache)# cache timeout active <minutes>
Router(config-flow-cache)# cache timeout inactive <seconds>
Router(config-flow-cache)# mask destination minimum <value>
Router(config-flow-cache)# mask source minimum <value>
Router(config-flow-cache)# export destination 10.10.10.10 1234
Router(config-flow-cache)# enabled
```

NetFlow Export Version 8 and Aggregation Configuration Example

```
Router # show ip flow export

  …
        Cache for <cache-type> aggregation:
           Exporting flows to 1.1.1.1 (9999)
           Exporting using source IP address 192.1.1.5
        1303631 flows exported in 332227 udp datagrams
           …
```

NetFlow Flow Keys on the Router

- By default, the flow keys are:
  - Source IP address, destination IP address, source port, destination port, layer 3 protocol type, TOS byte (DSCP), input interface
- The 12 NetFlow aggregation allows to reduce/change the number of flow keys
  - Example: source prefix aggregation = source network, source interface
  - Can be seen as a different view of the main cache
- Egress NetFlow, MPLS aware NetFlow, etc.
  - Will specify new flow keys
- Note: on the Cisco Catalyst®, we speak of the flow mask
  - Define the flow keys

Flow Keys on the Cisco Catalyst 6500/7600 the Flow Mask

Full-Interface					
VLAN	SRC IP	DST IP	IP Protocol	Src Port	Dst Port
Full					
VLAN	SRC IP	DST IP	IP Protocol	Src Port	Dst Port
Destination-Source-Interface					
VLAN	SRC IP	DST IP	IP Protocol	Src Port	Dst Port
Source-Only					
VLAN	SRC IP	DST IP	IP Protocol	Src Port	Dst Port
Destination-Only					
VLAN	SRC IP	DST IP	IP Protocol	Src Port	Dst Port
Destination-Source					
VLAN	SRC IP	DST IP	IP Protocol	Src Port	Dst Port

Flow Keys

NetFlow Version 9

Extensibility and Flexibility Requirements Phases Approach

- New requirements: build a flexible and extensible NetFlow
- Phase 1: NetFlow version 9, completed
  - Advantages: extensibility
    - Integrate new technologies/data types quicker (MPLS, IPv6, BGP next hop, etc.)
    - Integrate new aggregations quicker
  - Note: for now, the template definitions are fixed
- Phase 2: Flexible NetFlow, completed
  - Advantages: cache and export content flexibility
    - User selection of flow keys
    - User definition of the records

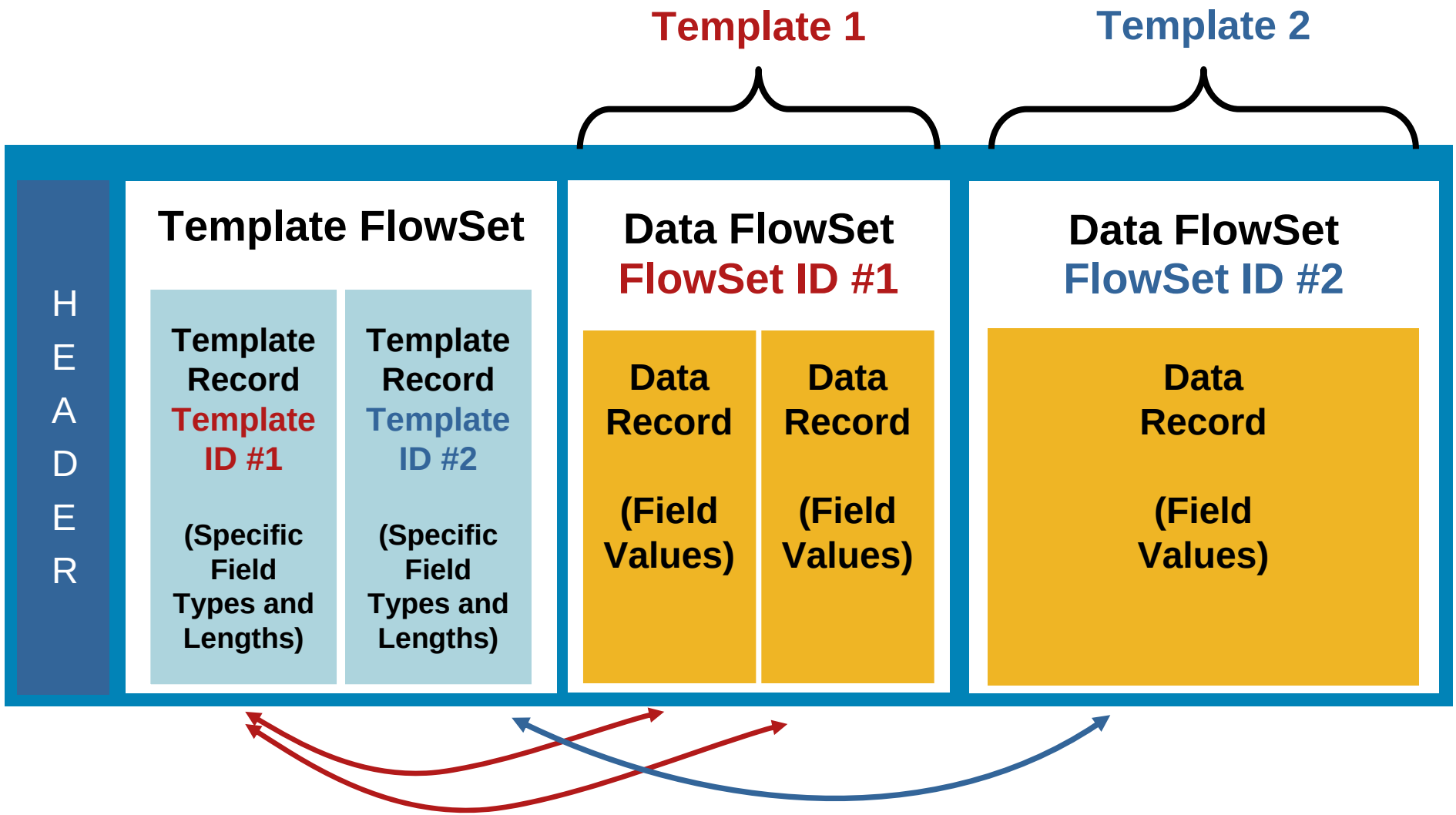
Exporting Process

Metering Process

NetFlow Version 9

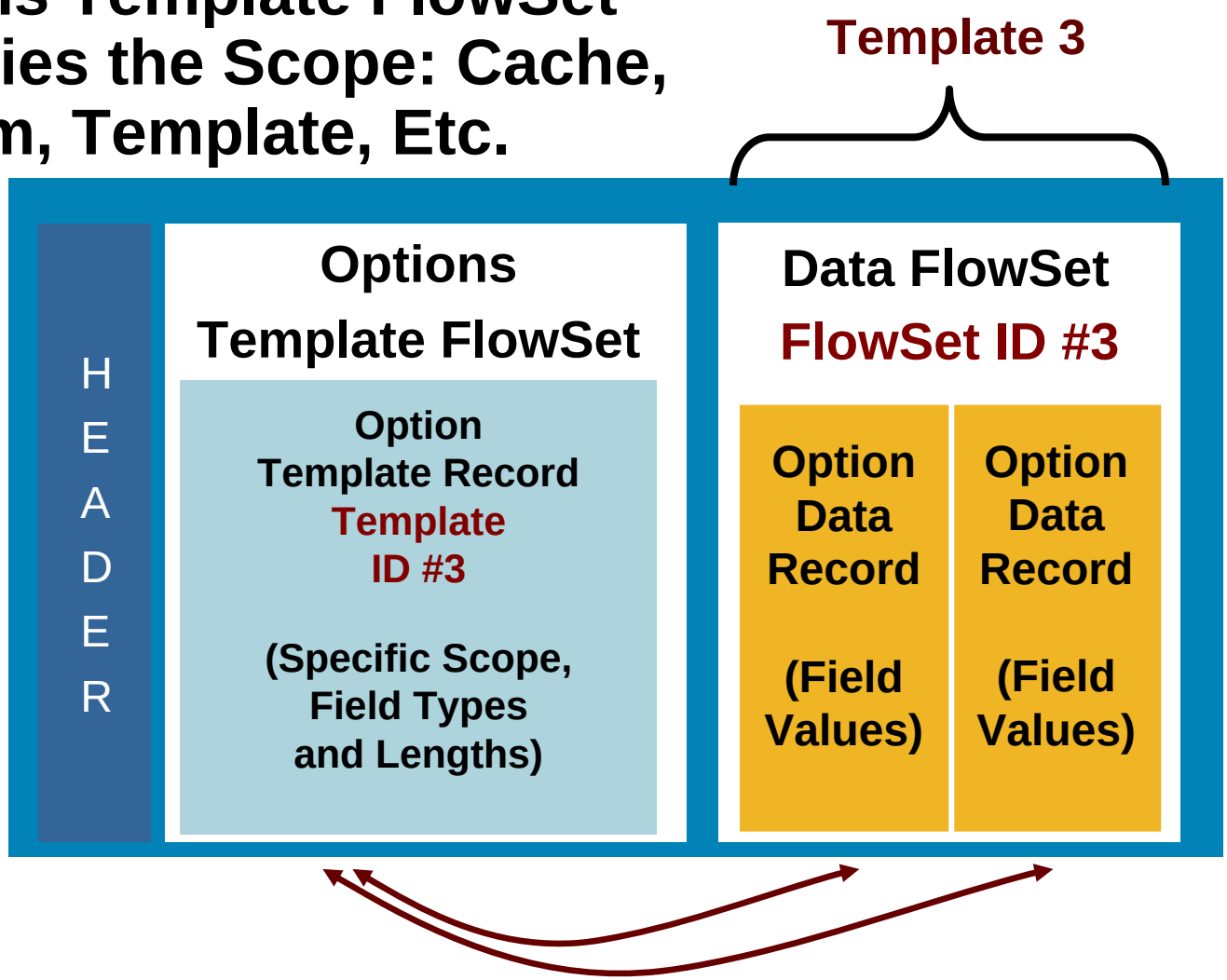
- Version 9 is an export protocol
  - No changes to the metering process
- Version 9 based on templates and separate flow records
  - Templates composed of type and length
  - Flow records composed of template ID and value
  - Sent the template regularly (configurable), because of UDP
- 800, 1700, 1800, 2600, 2800, 3600, 6500/7600, 7200, 7300, 7500, cat6000, 7600, 10000, 12000, CRS-1, etc.
  - 12.0(24)S, 12.3(1), 12.2(18)S
  - 12.2(18)SXF
  - 12.2(31)SB
  - Cisco IOS®-XR 3.2
- RFC3954 “Cisco Systems® NetFlow Services Export Version 9”

NetFlow Version 9 Export Packet



NetFlow Version 9 Export Packet

Options Template FlowSet Specifies the Scope: Cache, System, Template, Etc.



Interface Name Export with NetFlow Version 9

New

- NetFlow has been exporting the ifIndex
- Instead of the collector polling the ifName MIB variable for a specific ifIndex, the matching (ifIndex, ifName) is sent in an option data record
- Introduced in 12.4(4)T

```
Router(config)# ip flow-export interface-names
```

NetFlow Version 9
Main Cache Configuration

```
router(config)# ip flow-export version [5|9] [origin-as|peer-as]
                  [bgp-nexthop]
router(config)# ip flow-export template options export-stats
router(config)# ip flow-export template options timeout-rate 5
router(config)# ip flow-export template options refresh-rate 60
router(config)# ip flow-export template timeout-rate 5
router(config)# ip flow-export template refresh-rate 20
router(config)# ip flow-export destination 10.10.10.10 9996
```

(Options) Templates Sent Every 5 Minutes or 20 Packets

Should You Export from the Main Cache with NetFlow Version 5 or Version 9?

NetFlow Version 9 Aggregation Cache Configuration

```
router(config)# ip flow-aggregation cache bgp-nexthop-tos
router(config-flow-cache)# export destination 11.11.11.11 9999
  destination Specify the Destination IP address
  version configure aggregation cache export version
router(config-flow-cache)# export version ?
  9  Version 9 export format
router(config-flow-cache)# export version 9
router(config-flow-cache)# enabled
```

Sometimes Available: in This Case We Have Only Version 9. Why?

NetFlow Version 9 Monitoring

```
Router# show ip flow export template
   Template Options Flag = 0
   Total number of Templates added = 5
   Total active Templates = 3
   Flow Templates active = 3
   Flow Templates added = 5
   Option Templates active = 0
   Option Templates added = 0
   Template ager polls = 423903
   Option Template ager polls = 0
Main cache version 9 export is enabled
 Template export information
   Template timeout = 30
   Template refresh rate = 20
 Option export information
   Option timeout = 30
   Option refresh rate = 20
```

MIB: cnfTemplateTable

MIB cnfTemplateExportInfoTable

New Features

CISCO-NETFLOW-MIB

New

- Managed objects to configure:
  - Flow cache, interface, export, peer-as versus origin-as
  - Exception: no sampled NetFlow configuration
- Managed objects to monitor:
  - Packet size distribution, number of bytes exported per second, number of flows/UDP datagrams exported, number of template active, export statistics, protocol statistics, etc.
- Report the top flows → more on this later
- The CISCO-NETFLOW-MIB.my is not:
  - A replacement for the traditional method of exporting a flow cache
- Introduced in 12.2(25)S and 12.3(7)T on the software based routers (7500 and below)
- Note:
  - Don't forget the threshold mechanism with the RMON event/alarm or the EVENT-MIB

NetFlow MIB Monitoring

```
Router # show ip flow export
...
  479272 flows exported in 69204 udp datagrams
  0 flows failed due to lack of export packet
  3 export packets were sent up to process level
  0 export packets were dropped due to no fib
  0 export packets were dropped due to adjacency issues
  0 export packets were dropped due to fragmentation failures
  0 export packets were dropped due to encapsulation fixup failures
  0 export packets were dropped enqueuing for the RP
  0 export packets were dropped due to IPC rate limiting
```

cnfESRecordsExported

cnfESPktsExported

cnfESPktsFailed

cnfESPktsDropped

- The export rate ratecnfESExportRate

  Useful to estimate the required bandwidth

NetFlow MIB Monitoring

```
Router# show ip cache flow
IP packet size distribution (311656 total packets):
   1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
   .356 .316 .144 .115 .004 .003 .000 .007 .001 .000 .002 .017 .018 .009 .000

    512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
   .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

...
Protocol       Total    Flows   Packets Bytes  Packets Active(Sec) Idle(Sec)
--------      Flows     /Sec     /Flow  /Pkt     /Sec     /Flow     /Flow
TCP-Telnet       33      0.0        65    40      0.0      18.4      10.0
TCP-WWW           3      0.0         5    45      0.0       3.0       1.2
TCP-BGP        5343      0.0         2    47      0.0       5.1      11.1
TCP-other       411      0.0         2    48      0.0       1.0      10.9
UDP-other     98614      0.4         2    76      0.9       2.1      10.8
ICMP           9519      0.0         9    71      0.4      21.3      11.5
Total:       113923      0.5         2    73      1.4       3.8      10.9
```

cnfPSPacketSizeDistribution

cnfPSProtocolStatTable

Egress NetFlow Accounting

New

- The NetFlow egress support feature allows NetFlow accounting to be implemented for egress (outgoing) traffic on an interface or subinterface
- Locally generated traffic (traffic that is generated by the router on which the NetFlow egress support feature is configured) will not be counted
- The NetFlow egress feature captures NetFlow statistics for IP traffic only; MPLS statistics are not captured
- 12.3(11)T on the software based routers (7500 and below)

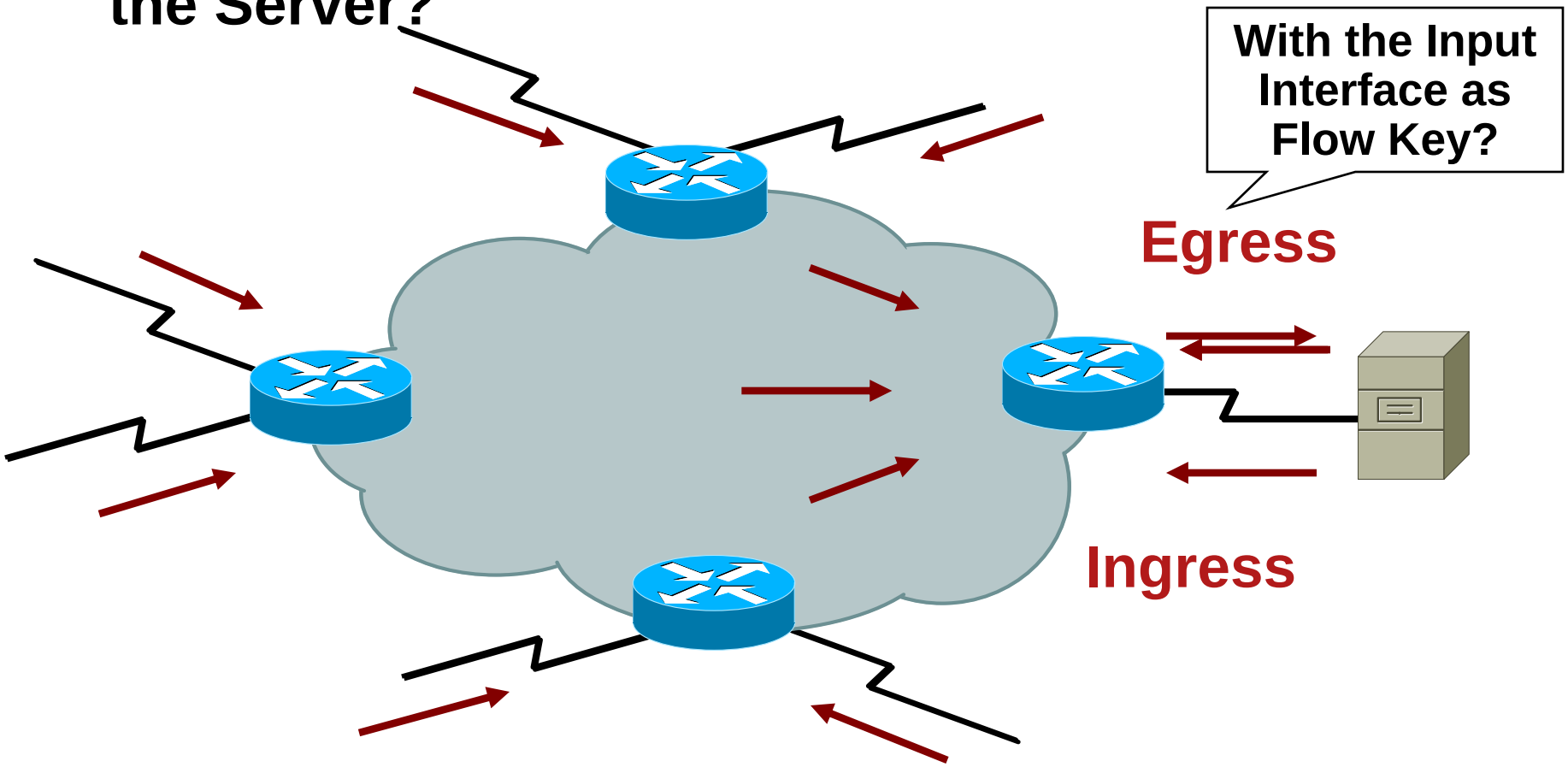
```
Router(config-if)# ip flow egress
```

Egress with the Input Interface as Flow Key

```
Router(config)# ip flow-egress input-interface
```

Egress NetFlow Accounting

How to Account the Traffic to/from the Server?



Attention to Double Count the Flow Records

Egress NetFlow Accounting

```
Router# show ip cache flow
…
SrcIf   SrcIPaddress DstIf   DstIPaddress   Pr SrcP DstP   Pkts
Et0/0   10.0.0.1     Et0/0*  10.0.1.1       01 0000 0000    5
Et0/1   10.0.0.2     Et0/1   10.0.1.2       01 0000 0000    5
```

The Asterisk (*) Indicates an Egress Flow

- Export the direction=egress with NetFlow version 9

NetFlow Reliable Export with SCTP
SCTP Introduction

New

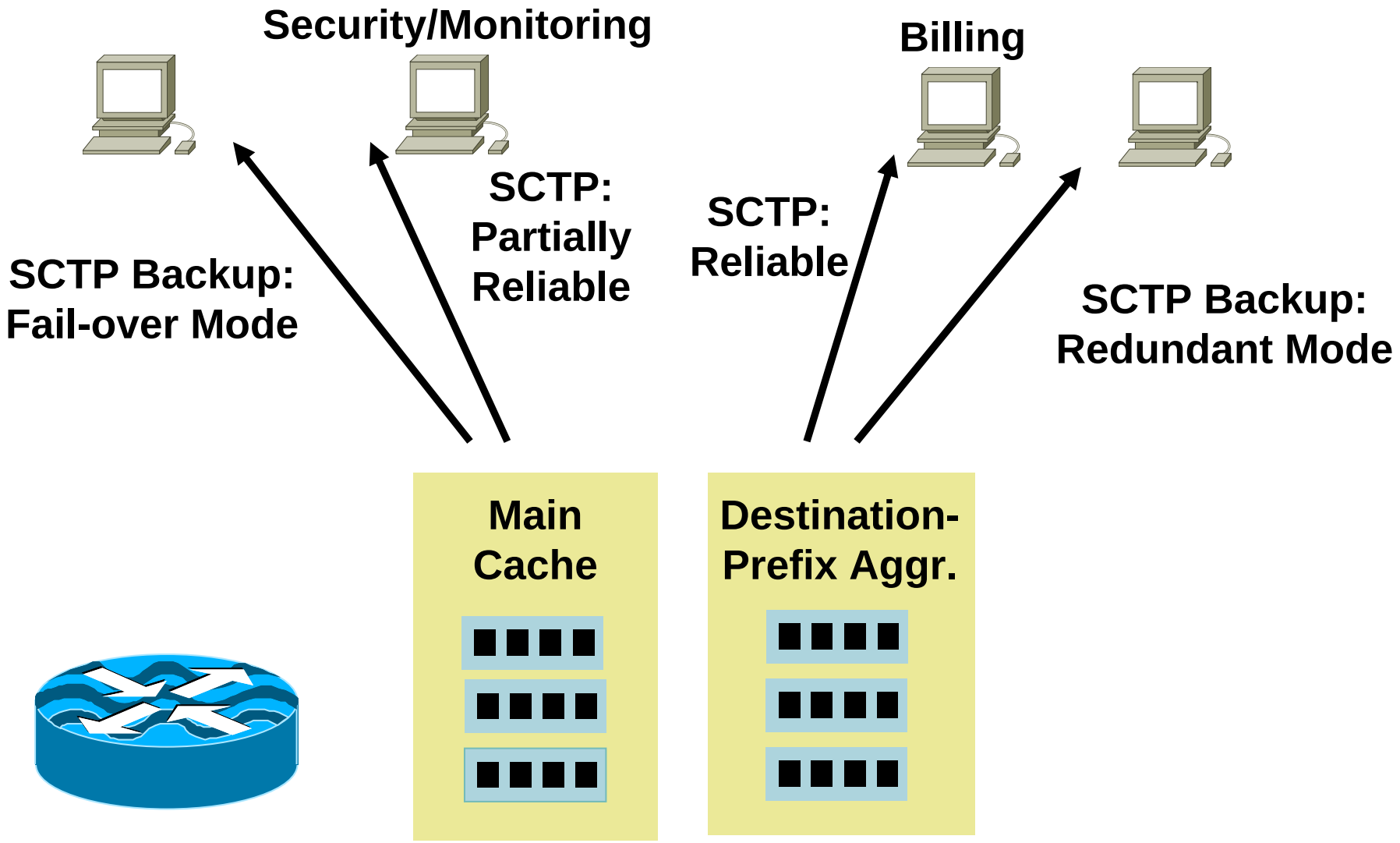
- UDP
  - Lack of security, congestion awareness, and reliability
  - However, speed and simplicity
- SCTP: stream control transport protocol (RFC2960)
  - Reliable data transfer
  - Congestion control and avoidance
  - Multihoming support
  - One association support for multi-streams
  - Security cookie against connection flood attack (SYN flood)
- SCTP-PR: SCTP partially reliable (RFC3578)
  - Three modes of reliability: reliable, partial reliable, unreliable
  - Each stream selects its mode of reliability

Note: "An Introduction to SCTP", RFC3286

NetFlow Reliable Export with SCTP

- SCTP-PR support for NetFlow version 5, 8, 9
- (Options) templates sent reliably
- Two primary SCTP export destinations (collectors) and two backup SCTP export destinations
  - For each cache: either main cache or aggregation cache(s)
- Backup
  - Fail-over mode: open the backup connection when the primary fails
  - Redundant mode: open the backup connection in advance, and already send the templates
  - Note that the backup inherits the reliability level from the primary
- 12.4(4)T on the software based routers (7500 and below)
- NetFlow collector SCTP support in version 6.0

Reliable Export with SCTP Example



Reliable Export with SCTP Example Configuration

```
Router(config)# ip flow-export destination 10.10.10.10 9999 sctp
Router(config-flow-export-sctp)# reliability partial buffer-limit 100
Router(config-flow-export-sctp)# backup destination 11.11.11.11 9999
Router(config-flow-export-sctp)# backup fail-over 1000
Router(config-flow-export-sctp)# backup mode fail-over

Router(config)# ip flow-aggregation cache destination-prefix
Router(config-flow-cache)# export destination 12.12.12.12 9999 sctp
Router(config-flow-export-sctp)# backup destination 13.13.13.13 9999
Router(config-flow-export-sctp)# backup mode redundant
Router(config-flow-export-sctp)# backup restore-time 1
Router(config-flow-export-sctp)# exit
Router(config-flow-cache)# enabled
```

Reliable Export with SCTP Example Show Command

```
Router# show ip flow export sctp verbose
IPv4 main cache exporting to 10.10.10.10, port 9999, partial
status: connected
backup mode: fail-over
104 flows exported in 84 sctp messages.
0 packets dropped due to lack of SCTP resources
fail-over time: 1000 milli-seconds
restore time:   25 seconds
backup: 11.11.11.11, port 9999
   status: not connected
   fail-overs: 0
   0 flows exported in 0 sctp messages.
   0 packets dropped due to lack of SCTP resources
destination-prefix cache exporting to 12.12.12.12, port 9999, full
status: connected
backup mode: redundant
57 flows exported in 42 sctp messages.
0 packets dropped due to lack of SCTP resources
fail-over time: 25 milli-seconds
restore time:   1 seconds
backup: 13.13.13.13, port 9999
   status: connected
   fail-overs: 0
   0 flows exported in 0 sctp messages.
   0 packets dropped due to lack of SCTP resources
```


Multicast NetFlow

New

- Multicast NetFlow ingress

  One flow with the replicated number of packets/bytes

```
Router(config-if)# ip multicast netflow ingress
```

- Multicast NetFlow egress

  One per outgoing interface, with the nonreplicated number of packets/bytes

```
Router(config-if)# ip multicast netflow egress
```

- Deduced the replication factor, multicast data that fails the RPF check
- No NetFlow export over multicast
- 12.0(27)S, 12.2(18)S, 12.3(1), 12.2(18)SXF

NetFlow Enabled Interfaces

New

```
Router# show ip flow interface
Serial0/0
  ip route-cache flow
Serial0/0.1
  ip flow egress
Serial0/3
  ip route-cache flow
FastEthernet1/0
  ip flow ingress
  flow-sampler benoit egress
```

Introduced in 12.3(7)T on the software based routers (7500 and below)

NetFlow VRF Export

New

- Allow the export of flow records within a VRF
- Valid for both SCTP and UDP export

```
Router(config)# ip flow-export destination 10.10.10.10 9999 vrf benoit <sctp|udp>
```

```
Router(config-flow-cache)#export destination 10.10.10.10 9999 vrf benoit <sctp|udp>
```

- Introduced in 12.4(4)T on the software based routers (7500 and below)

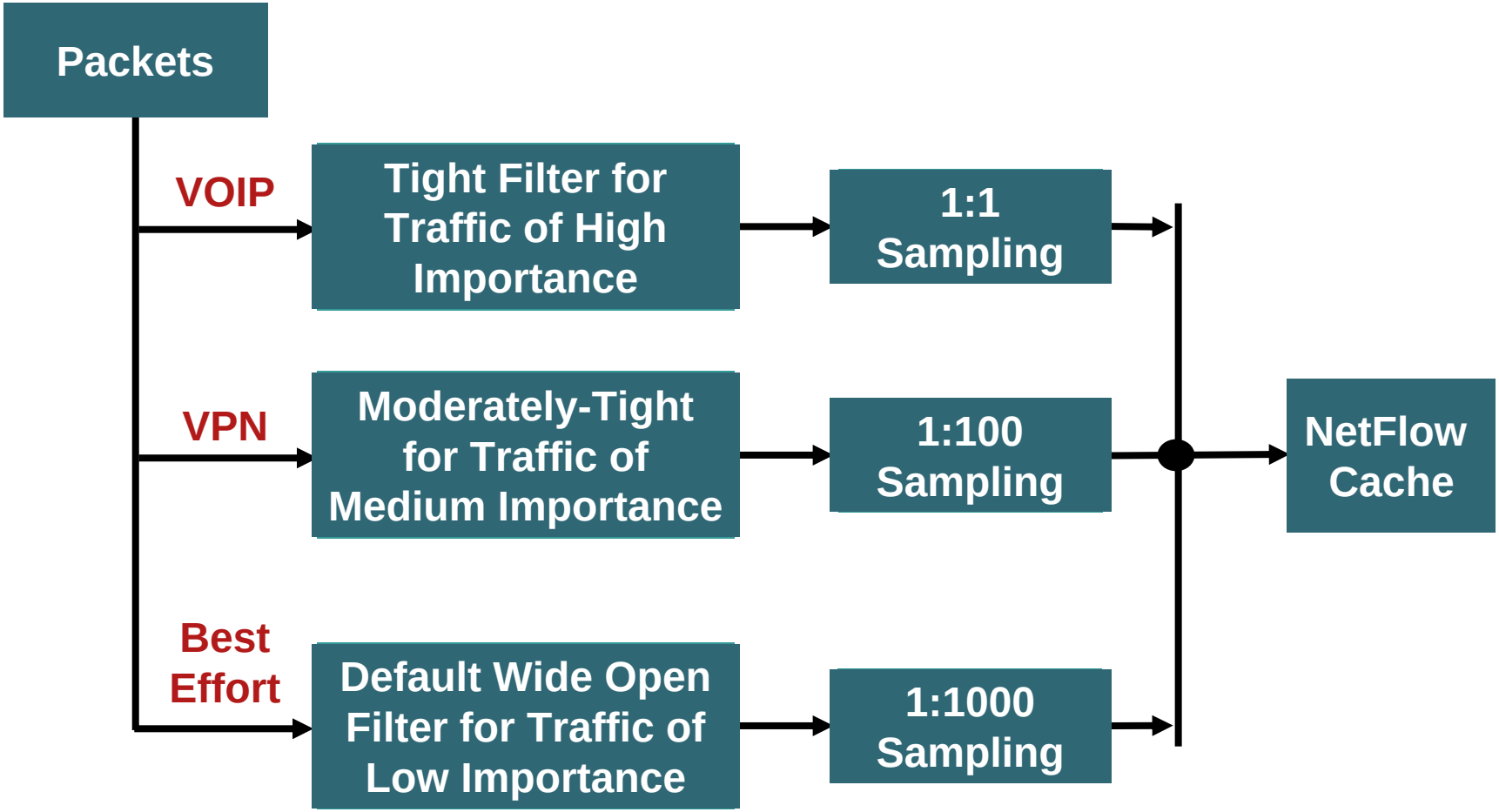
NetFlow and IPv6

New

- Monitors the IPv6 traffic
- Based on NetFlow version 9
- For both ingress and egress traffic
- Non-sampled
- No NetFlow export over IPv6; still IPv4
- All configuration is the same: replace “ip” by “ipv6”
- 12.3(4)T, 12.2(25)S for on the software based routers (7500 and below)
- 1st half 2007 for the Cisco Catalyst 6500/7600

NetFlow Input Filters Example

New



NetFlow Input Filters

- Support prefiltering for traffic for NetFlow processing
- Modular QoS command line (MQC) will provide the filtering mechanism for NetFlow
  - Classification by IP source and destination addresses, layer 4 protocol and port numbers, Incoming interface, MAC address, DSCP
  - Layer 2 information such as Frame Relay DE bits, Ethernet 802.1p bits
  - Network based application recognition (NBAR)
- Ability to sample filtered data at different rates, depending on how interesting the traffic is
- 12.3(4)T, 12.2(25)S

NetFlow Input Filters Configuration

```
Router(config)# class-map high_importance_class
Router(config-cmap)# match access-group 101
Router(config-cmap)# exit
Router(config)# class-map medium_importance_class
Router(config-cmap)# match access-group 102
Router(config-cmap)# exit
```

Define Traffic Classes (MQC)

```
Router(config)# flow-sampler-map high_sampling
Router(config-sampler-map)# mode random one-out-of 1
Router(config-sampler-map)# exit
Router(config)# flow-sampler-map medium_sampling
Router(config-sampler-map)# mode random one-out-of 100
Router(config-sampler-map)# exit
Router(config)# flow-sampler-map low_sampling
Router(config-sampler-map)# mode random one-out-of 1000
Router(config-sampler-map)# exit 18
```

Define NetFlow Samplers

NetFlow Input Filters Configuration

```
Router(config)# policy-map mypolicy
Router(config-pmap)# class high_importance_class
Router(config-pmap-c)# flow-sampler high_sampling
Router(config-pmap-c)# exit
Router(config-pmap)# class medium_importance_class
Router(config-pmap-c)# flow-sampler medium_sampling
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap-c)# flow-sampler low_sampling
Router(config-pmap-c)# exit
```

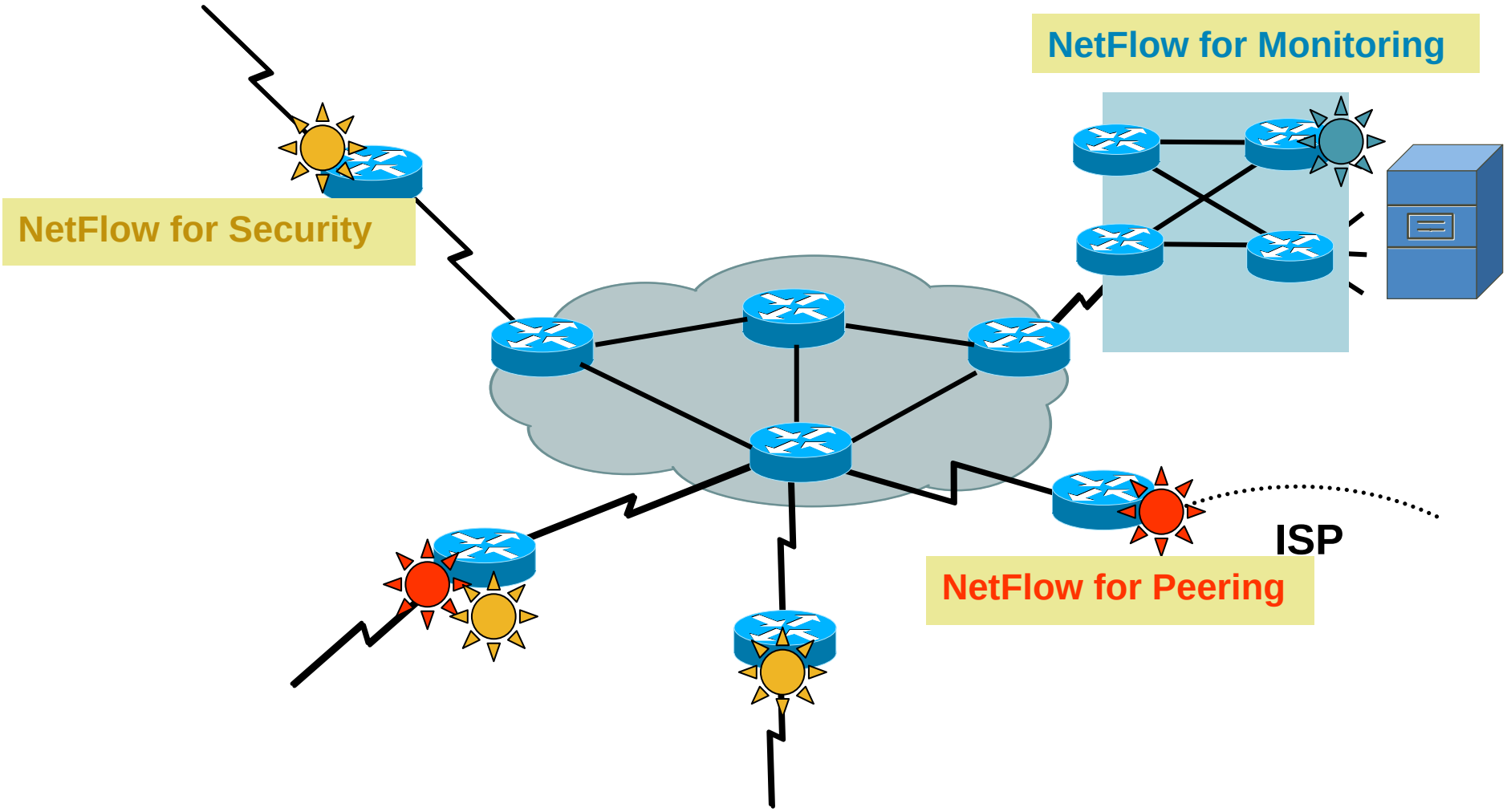
Define Policy with NetFlow Sampling Actions

```
Router(config)# interface POS1/0
Router(config-if)# service-policy input mypolicy
Router(config-if)# exit
Router(config)# interface ATM2/0
Router(config-if)# service-policy input mypolicy
```

Applying Policy with NextFlow Sampling Actions to Interface

Flexible NetFlow

Typical NetFlow Deployment



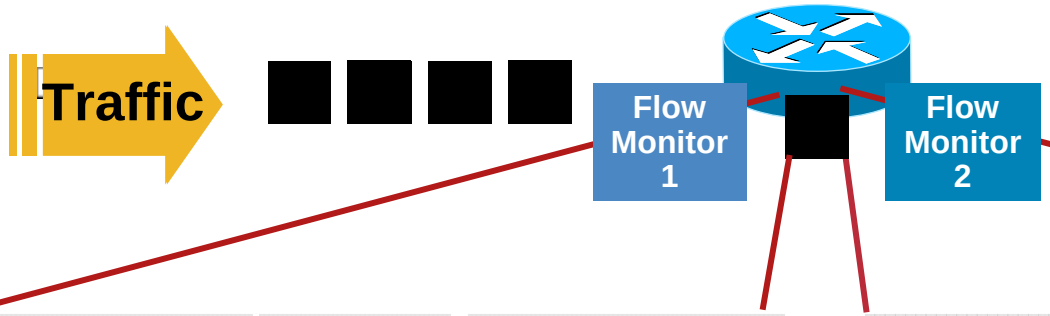
Flexible NetFlow
High Level Concepts and Advantages

- Flexible NetFlow feature allows user configurable NetFlow record formats, selecting from a collection of fields:
  - Key
  - Non-key
  - Counter
  - Timestamp
- Advantages:
  - Tailor a cache for specific applications, not covered by existing 21 NetFlow features
  - Better scalability since flow record customization for particular application reduces number of flows to monitor
  - Different NetFlow configuration:
    - Per subinterface
    - Per direction (ingress/egress)
    - Per sampler
    - Etc.

Flow Key and Non-Key Fields

- Choice of flow keys includes IPv4 header, transport (TCP, UDP), routing, flow (direction, sampler), interface
- Non-key fields are not used to define a flow and are exported along with the flow and provide additional information
  - Traditional IP NF non-key fields:
    - Source and destination AS's
    - Source and destination IP prefix masks
    - IP address of next hop router
    - TCP flags
    - Output interface
    - Note: given by the value of the first packet of the flow
  - NF features provide per flow statistics:
    - Number of packets and bytes in flow
    - Timestamps for first and last packets in flow

Flexible NetFlow
Multiple Monitors with Unique Key Fields



Key Fields	Packet 1
Source IP	3.3.3.3
Destination IP	2.2.2.2
Source Port	23
Destination Oort	22078
Layer 3 Protocol	TCP - 6
TOS Byte	0
Input Interface	Ethernet 0

Non-Key Fields
Packets
Bytes
Timestamps
Next Hop Address

Key Fields	Packet 1
Source IP	3.3.3.3
Dest IP	2.2.2.2
Input Interface	Ethernet 0
SYN Flag	0

Non-Key Fields
Packets
Timestamps

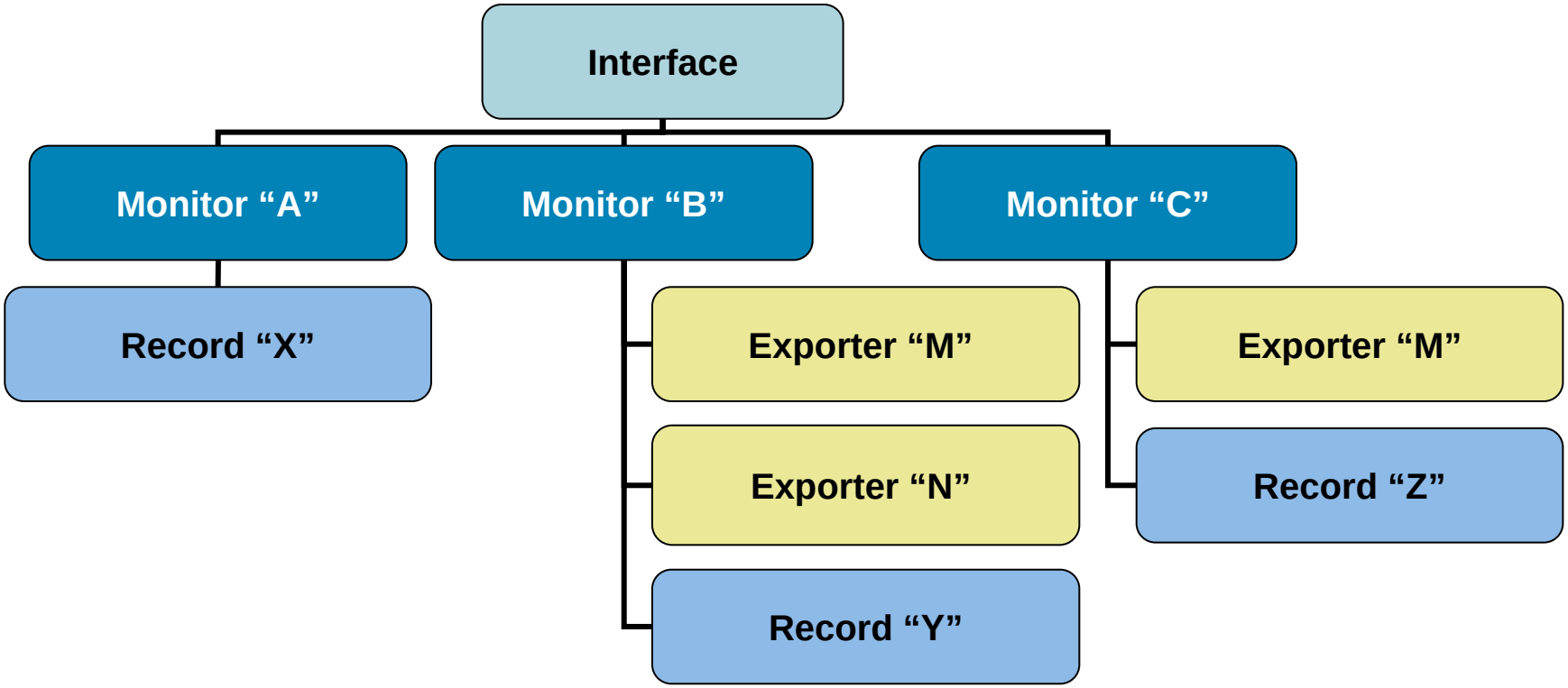
Traffic Analysis Cache

Source IP	Dest. IP	Source Port	Dest. Port	Protocol	TOS	Input I/F	…	Pkts
3.3.3.3	2.2.2.2	23	22078	6	0	E0	…	1100

Security Analysis Cache

Source IP	Dest. IP	Input I/F	Flag	…	Pkts
3.3.3.3	2.2.2.2	E0	0	…	11000

Flexible NetFlow Model



- A single record per monitor
- Potentially multiple monitors per interface
- Potentially multiple exporters per monitor

Flexible NetFlow Components

- Flow record—defines what is captured by NetFlow
  - Two kinds of flow records: predefined or user-defined
  - Include key and non-key fields
- Flow exporter—where NetFlow will be exported
  - Multiple flow exporters per flow monitor
- Flow monitor—a flow cache containing flow records
  - Cache creation for a specific flow record
  - Applied to an interface
  - Bound to one or more flow exporter(s)
  - Packet sampling possible per flow monitor

Flexible NetFlow Configuration

Configure the Exporter

- Where do I want my data sent?

Configure the Flow Record

- What data do I want to meter?

Configure the Flow Monitor

- Creates a new NetFlow cache
- Attach the flow record
- Exporter is attached to the cache
- Potential sampling configuration

Configure the Interface

- Configure NetFlow on the interface

Predefined Record for Traditional NetFlow

Configure the Exporter

```
Router(config)#flow exporter my-exporter-server
Router(config-flow-exporter)#destination 1.1.1.1
```

Configure the Flow Record

Not necessary for predefined types

Configure the Flow Monitor

```
Router(config)#flow monitor my-monitor
Router(config-flow-monitor)#exporter my-exporter-server
Router(config-flow-monitor)#record netflow original-input
```

Configure the Interface

```
Router(config)#int s3/0
Router(config-if)#ip flow monitor my-monitor input
```

Predefined Record for Traditional NetFlow

- All aggregations are possible, for quick backwards compatibility

```
Router(config)# flow monitor my-monitor
Router(config-flow-monitor)# record netflow ipv4 ?
  as                        AS aggregation schemes
  as-tos                    AS and TOS aggregation schemes
  bgp-nexthop-tos           BGP next-hop and TOS aggregation schemes
  destination-prefix        Destination Prefix aggregation schemes
  destination-prefix-tos    Destination Prefix and TOS aggregation schemes
  original-input            Traditional IPv4 input NetFlow
  original-output           Traditional IPv4 output NetFlow
  prefix                    Source and Destination Prefixes aggregation schemes
  prefix-port               Prefixes and Ports aggregation scheme
  prefix-tos                Prefixes and TOS aggregation schemes
  protocol-port             Protocol and Ports aggregation scheme
  protocol-port-tos         Protocol, Ports and TOS aggregation scheme
  source-prefix             Source AS and Prefix aggregation schemes
  source-prefix-tos         Source Prefix and TOS aggregation schemes
```

Configure a User-Defined Flow Record

Configure the Exporter

```
Router(config)#flow exporter my-exporter

Router(config-flow-exporter)#destination 1.1.1.1
```

Configure the Flow Record

```
Router(config)#flow record my-record
Router(config-flow-record)#match ipv4 destination address
Router(config-flow-record)#match ipv4 source address
Router(config-flow-record)#collect counter bytes
```

Configure the Flow Monitor

```
Router(config)#flow monitor my-monitor

Router(config-flow-monitor)#exporter my-exporter

Router(config-flow-monitor)#record my-record
```

Configure the Interface

```
Router(config)#int s3/0

Router(config-if)#ip flow monitor my-monitor input
```

Flexible NetFlow
User Defined Record Configuration

```
Router(config)# flow record my-record
Router(config-flow-record)# match     -> Specify a key field
Router(config-flow-record)# collect   -> Specify a non-key field

Router(config-flow-record)# match ?
      flow            Flow identifying fields
      interface       Interface fields
      ipv4            IPv4 fields
      routing         routing attributes
      transport       Transport layer field

Router(config-flow-record)# collect ?
      counter     Counter fields
      flow        Flow identifying fields
      interface   Interface fields
      ipv4        IPv4 fields
      routing     IPv4 routing attributes
      timestamp   Timestamp fields
      transport   Transport layer fields
```

Flexible Flow Record—Key Fields

IPv4	
IP (Source or Destination)	Payload Size
Prefix (Source or Destination)	Packet Section (Header)
Mask (Source or Destination)	Packet Section (Payload)
Minimum-Mask (Source or Destination)	TTL
Protocol	Options bitmap
Fragmentation Flags	Version
Fragmentation Offset	Precedence
ID	DSCP
Header Length	TOS
Total Length	

Routing
src or dest AS
Peer AS
Traffic Index
Forwarding Status
Is-Multicast
IGP Next Hop
BGP Next Hop

Flow
Sampler ID
Direction

Transport	
Destination Port	TCP Flag: ACK
Source Port	TCP Flag: CWR
ICMP Code	TCP Flag: ECE
ICMP Type	TCP Flag: FIN
IGMP Type	TCP Flag: PSH
TCP ACK Number	TCP Flag: RST
TCP Header Length	TCP Flag: SYN
TCP Sequence Number	TCP Flag: URG
TCP Window-Size	UDP Message Length
TCP Source Port	UDP Source Port
TCP Destination Port	UDP Destination Port
TCP Urgent Pointer	

Flexible Flow Record—Non-Key Fields

Counters
Bytes
Bytes Long
Bytes Square Sum
Bytes Square Sum Long
Packets
Packets Long

Timestamp
sysUpTime First Packet
sysUpTime First Packet

IPv4
Total Length Minimum
Total Length Maximum
TTL Minimum
TTL Maximum

- Plus any of the potential “key” field: will be the value from the first packet in the flow

Flow Exporter Configuration

```
flow exporter <exporter-name>
     destination <ipv4-address> [vrf <vrf-name>]
     dscp <value>
     option {exporter-stats | interface-table | sampler-table}
       timeout <value in sec>
     output-features
     source <interface-name>
     template resend timeout <value in sec>
     transport udp <destination-port>
     ttl <value>
```

3 Types of Options Data Record

Will Take the Output Features: QoS, IPSec, Etc.

(Option) Template Sent Every X Seconds

- Later phase: IPFIX, SCTP, IPv6 export

Flexible Monitor Configuration

Potentially Multiple

3 Types of Cache:
See Next Slides

```
flow monitor <monitor-name>
  record <record-name>
  exporter <exporter-name>
  cache type {normal | immediate | permanent}
  cache entries <number-of-entries>
  cache timeout {active | inactive | update} <value-in-sec>
  statistics packet protocol
  statistics packet size
```

Collect Size
Distribution Statistics

Collect Protocol
Distribution Statistics

Three Types of NetFlow Caches

- Normal cache
  - Similar to today's NetFlow
  - More flexible active and inactive timers: one second minimum
- Immediate cache
  - Flow accounts for a single packet
  - Desirable for real-time traffic monitoring, DDoS detection, logging
  - Desirable when only very small flows are expected (ex: sampling)
  - Caution: may result in a large amount of export data
- Permanent cache
  - To track a set of flows without expiring the flows from the cache
  - Entire cache is periodically exported (update timer)
  - After the cache is full (size configurable), new flows will not be monitored
  - Uses update counters rather than delta counters

Complete Permanent Flexible NetFlow Configuration Example

- Per DSCP accounting flow record definition:

```
Router(config)# flow record my-dscp-record
Router(config-flow-record)# match ipv4 dscp
Router(config-flow-record)# match interface input
Router(config-flow-record)# collect counter bytes long
Router(config-flow-record)# collect counter packets long

Router(config)# flow monitor my-dscp-monitor
Router(config-flow-record)# description dscp:bytes and packets
Router(config-flow-record)# record my-dscp-record
Router(config-flow-record)# cache type permanent
Router(config-flow-record)# cache entries 256

Router(config)# interface GigabitEthernet 0/1
Router(config)# ip flow monitor my-dscp-monitor input
```

64 Bit Counter

- This would replace “IP accounting precedence”

Complete Permanent Flexible NetFlow Configuration Example

Extra Options:
Csv, Table, Record

```
Router#show flow monitor my-dscp-monitor cache
  Cache type:                                Permanent
  Cache size:                                      256
  Current entries:                                   0
  High Watermark:                                    0

  Flows added:                                       0
  Updates sent          (  1800 secs)                0

IP DSCP  INTF INPUT      bytes long perm    pkts long perm
=======  ============  ====================  ================

 0x00     Gi0/1                   1000                  10
 0x01     Gi0/1                    500                   5
```

Flow Keys in Upper Case

Flexible NetFlow Activation on Interface

Send the "sampler-table" Option

```
Router(config-if)# ip flow monitor <monitor-name>
                                [sampler <sampler-name>]
                               [input | output]
```

For the Input or Output Traffic.
Does Not Determine the Flow Key

- Deterministic or random is available

```
Router(config)# sampler <sampler-name>
mode [deterministic | random] <value N> out-of <value M>
```

Packet Section Fields

- Contiguous chunk of a packet of a user configurable size, used as a key or a non-key field
- Sections used for detailed traffic monitoring, DDoS attack investigation, worm detection, other security applications
- Chunk defined as flow key, should be used in sampled mode with immediate aging cache
- Starts at the beginning of the IPv4 header

```
collect or match ipv4 header <size in bytes>
```

- Immediately follows the IPv4 header

```
collect or match ipv4 payload <size in bytes>
```

Useful Show Commands

- List of all possible information elements

  “show flow exporter export-ids netflow-v9”

- Template assignment

  “show flow exporter template”

- High watermark in the cache

  “show flow monitor <flow-monitor> statistics

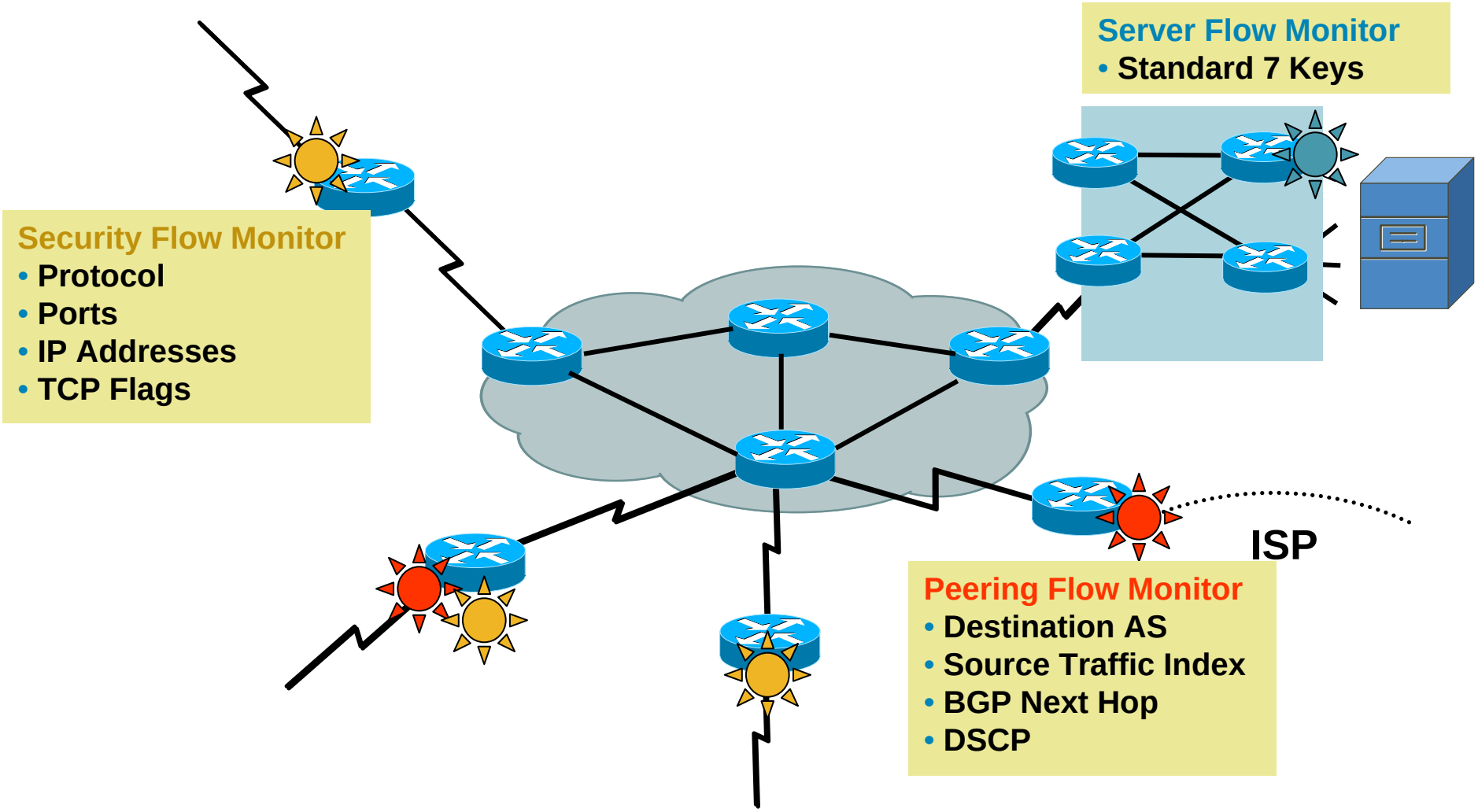
- NetFlow configuration

  “show running flow [exporter | monitor | record]

- Cache collisions

  “show flow monitor my-monitor internal”

Deployment Example



Flexible NetFlow Support

- Platforms:
  - 800, 1700, 1800, 2600, 2800, 3700, 3800, 7200, 7301: 12.4(9)T
  - 12000: 12.0(33)S, future
- CEF or DCEF required
- Version 9 is the only export format supported
  - IPFIX in the future
- Current NetFlow features not supported
  - MIB
  - Top talkers
  - IPv6 in the future

NetFlow for Security

What Does a DoS Attack Look Like?

```
Router# show ip cache flow
…
SrcIf   SrcIPaddress SrcP  SrcAS  DstIf DstIPaddress  DstP  DstAS   Pr Pkts B/Pk
29      192.1.6.69     77    aaa     49   194.20.2.2  1308   bbb    6    1   40
29      192.1.6.222  1243    aaa     49   194.20.2.2  1774   bbb    6    1   40
29      192.1.6.108  1076    aaa     49   194.20.2.2  1869   bbb    6    1   40
29      192.1.6.159   903    aaa     49   194.20.2.2  1050   bbb    6    1   40
29      192.1.6.54    730    aaa     49   194.20.2.2  2018   bbb    6    1   40
29      192.1.6.136   559    aaa     49   194.20.2.2  1821   bbb    6    1   40
29      192.1.6.216   383    aaa     49   194.20.2.2  1516   bbb    6    1   40
29      192.1.6.111    45    aaa     49   194.20.2.2  1894   bbb    6    1   40
29      192.1.6.29   1209    aaa     49   194.20.2.2  1600   bbb    6    1   40
```

- Typical DoS attacks have the same (or similar) entries:

  Input interface, destination IP, 1 packet per flow, constant bytes per packet (B/Pk)

- Don't forget "show ip cache verbose flow | include …"
- Export to a security oriented collector: CS-MARS, Arbor collector

NetFlow L2 and Security Monitoring

New

- Layer 2 IP header fields
  - Source MAC address field from frames that are received by the NetFlow router
  - Destination MAC address field from frames that are transmitted by the NetFlow router
  - Received VLAN ID field (802.1q and Cisco's ISL)
  - Transmitted VLAN ID field (802.1q and Cisco's ISL)
- Extra layer 3 IP header fields
  - Time-to-live field
  - Identification field
  - Packet length field
  - ICMP type and code
  - Fragment offset
- Targeted for security: to help identify network attacks and their origin
- For IPv4 and IPv6
- Introduced in 12.3(14)T on the software based routers (7500 and below)
  - Fragment offset introduced in 12.4(2)T

NetFlow L2 and Security Monitoring L3 Packet Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|  IHL  |Type of Service|          Total Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification        |Flags|      Fragment Offset    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live |    Protocol   |         Header Checksum       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination Address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

NetFlow L2 and Security Monitoring
Current NetFlow L3 Fields

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|  IHL  |Type of Service|          Total Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification        |Flags|      Fragment Offset    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live |    Protocol   |         Header Checksum       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination Address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

NetFlow L2 and Security Monitoring
Extra NetFlow L3 Fields

Attacks That Use Consistent Packet Size or Worms That Use Consistent Packet Size

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|  IHL  |Type of Service|          Total Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification        |Flags|      Fragment Offset    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live |    Protocol   |         Header Checksum       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                       Source Address
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                    Destination Address
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Flow Issued From the Same Origin

Several Flows with the Same Fragment Offset: Same Packet Sent over and over

Very Large Packets or Attacks That Might Always Have the Same Generated Identification

NetFlow L2 and Security Monitoring

```
Router(config)# ip flow-capture icmp
Router(config)# ip flow-capture ip-id
Router(config)# ip flow-capture mac-addresses
Router(config)# ip flow-capture packet-length
Router(config)# ip flow-capture ttl
Router(config)# ip flow-capture vlan-id
Router(config)# ip flow-capture fragment-offset
```

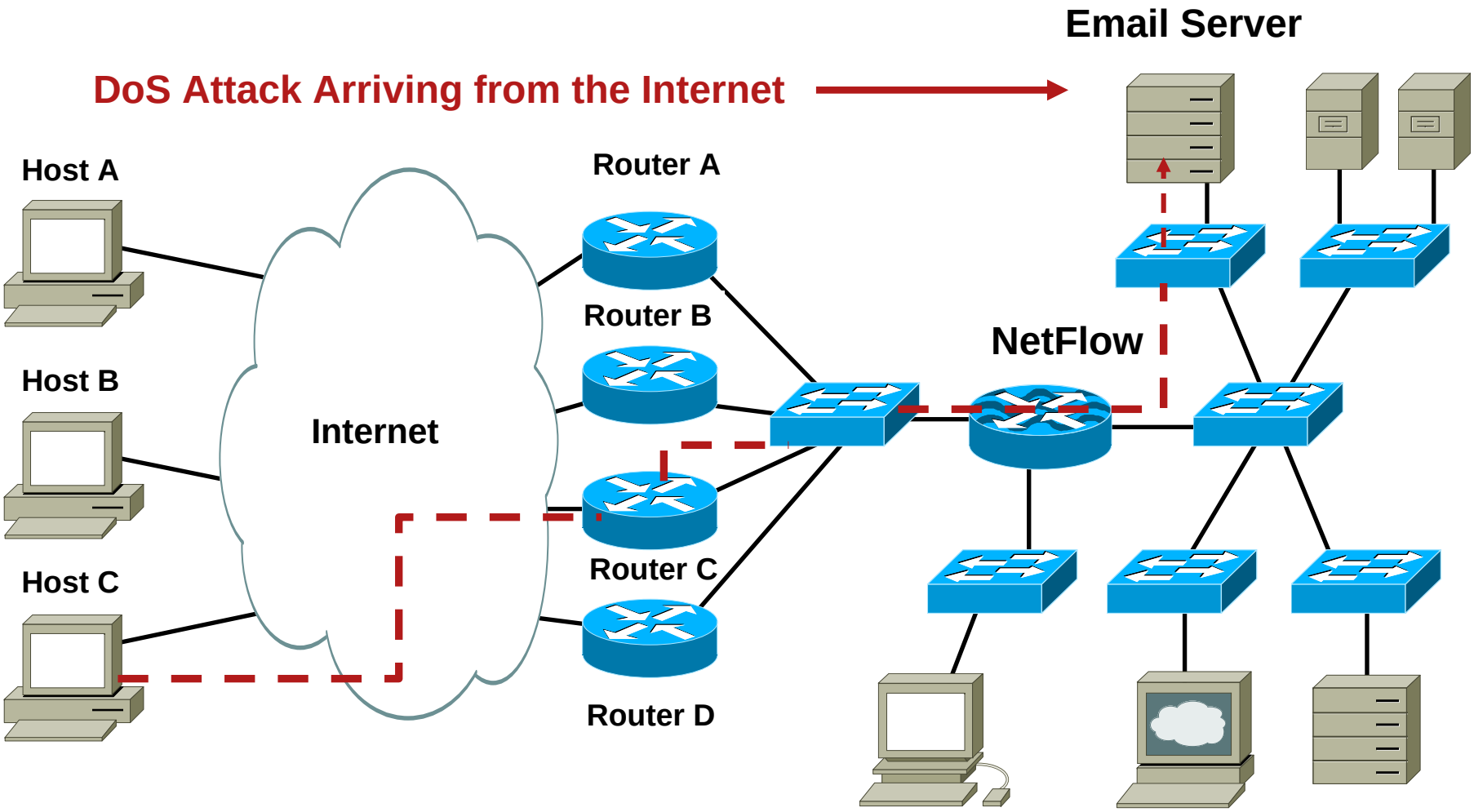
- Not flow keys, the value of the first packet of the flow
  - Exception for packet length: min/max
  - Exception for the TTL: min/max
  - Fragment-offset: the first fragmented packet
- Complete the main cache, not the aggregation caches
  - Info lost if an aggregation cache is used
- Currently not available with the MIB

NetFlow L2 and Security Monitoring

```
Router# show ip cache verbose flow
…
SrcIf          SrcIPaddress    DstIf          DstIPaddress    Pr TOS Flgs  Pkts
Port Msk AS                    Port Msk AS    NextHop              B/Pk  Active

Et0/0.1        10.251.138.218  Et1/0.1        172.16.10.2     06 80  00      65
0015 /0  0                     0015 /0  0     0.0.0.0               840    10.8
MAC: (VLAN id) aaaa.bbbb.cc03  (005)          aaaa.bbbb.cc06  (006)
Min plen:      840                            Max plen:       840
Min TTL:        59                            Max TTL:         59
IP id:           0
```

One Flow Entry

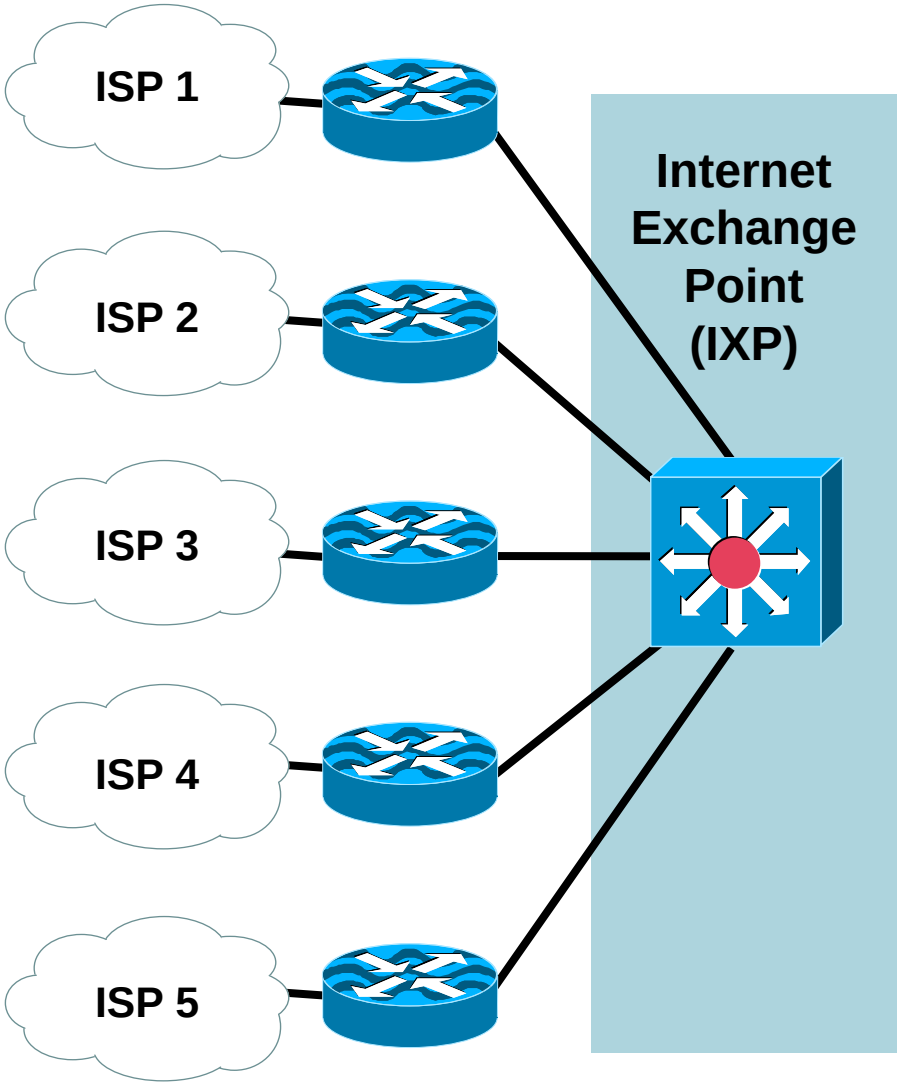
NetFlow L2 and Security Monitoring Source MAC Address



Report the MAC Address for Ethernet, FastEthernet, and GigabitEthernet

NetFlow L2 and Security Monitoring Internet eXchange Point

- Internet exchange points require the accounting per MAC address:
  - Incoming
  - Outgoing
- NetFlow solution is more granular than the “IP accounting MAC address” feature



NetFlow Top Talkers

New

- The flows that are generating the heaviest traffic in the cache are known as the "top talkers". Prefer “top flows”
- Allows flows to be sorted by either of the following criteria:
  - By the total number of packets in each top talker
  - By the total number of bytes in each top talker
- Match criteria for the top talkers, work like a filter
- The top talkers can be retrieved via the CISCO-NETFLOW-MIB (cnfTopFlowsTable)
- A new separate cache
  - Similar output of the show ip cache flow or show ip cache verbose flow command
  - Generated on the fly
  - Frozen for the “cache-timeout” value
- Introduced in 12.2(25)S and 12.3(11)T on the software based routers (7500 and below)

NetFlow Top Talkers Example 1

```
Router(config)# ip flow-top-talkers
Router(config-flow-top-talkers)# top 50
Router(config-flow-top-talkers)# sort-by <packets | bytes>
Router(config-flow-top-talkers)# cache-timeout 2000
```

```
Router# show ip flow top-talkers verbose

SrcIf          SrcIPaddress    DstIf          DstIPaddress    Pr TOS Flgs  Pkts
Port Msk AS                    Port Msk AS    NextHop              B/Pk  Active
IPM: OPkts    OBytes
Fa1/0          10.48.71.9      Local          10.48.71.9      01 C0  10      56
0000 /24 0                     0303 /24 0     0.0.0.0                56   171.0
ICMP type:       3                            ICMP code:        3

Se0/0          192.1.1.97      Se0/3          192.1.1.110     01 00  00      12
0000 /30 0                     0000 /30 0     192.1.1.108          1436     2.8
ICMP type:       0                            ICMP code:        0
```

NetFlow Top Talkers Example 2

```
Router(config)# ip flow-top-talkers
Router(config-flow-top-talkers)# top 50
Router(config-flow-top-talkers)# sort-by packets
Router(config-flow-top-talkers)# cache-timeout 2000
Router(config-flow-top-talkers)# match source address 192.1.1.97/32
Router(config-flow-top-talkers)# match destination address 192.1.1.110/32
```

```
Router# show ip flow top-talkers verbose

SrcIf          SrcIPaddress    DstIf          DstIPaddress    Pr TOS Flgs  Pkts
Port Msk AS                    Port Msk AS    NextHop              B/Pk  Active

Se0/0          192.1.1.97      Se0/3          192.1.1.110     01 00  00     12
0000 /30 0                     0000 /30 0     192.1.1.108          1436    2.8
ICMP type:       0                            ICMP code:        0
```

NetFlow Top Talkers Example 2

```
Router(config)# ip flow-top-talkers
Router(config-flow-top-talkers)# top 50
Router(config-flow-top-talkers)# sort-by packets
Router(config-flow-top-talkers)# cache-timeout 2000
Router(config-flow-top-talkers)# match source address 192.1.1.97/32
Router(config-flow-top-talkers)# match destination address 192.1.1.110/32
```

match [[source address | destination address | nexthop address] [*ip-address*] [*mask* | */nn*]] [[source port | destination port] [*port-number* | min *port* | max *port* | min *port* max *port*]] [[source as | destination as] *as-number*] [[input-interface | output-interface] *interface*] [tos [*tos-value* | dscp *dscp-value* | precedence *precedence-value*]] [protocol [*protocol-number* | tcp | udp]] [flow-sampler *flow-sampler-name*] [class-map *class*] [packet-range | byte-range [[*min-range-number* *max-range-number*] [min *minimum-range* | max *maximum-range* | min *minimum-range* max *maximum-range*]]]

Egress NetFlow and Top Talkers

```
Router(config)# ip flow-top-talkers
Router(config-flow-top-talkers)# match source address 192.1.1.97/32
Router(config-flow-top-talkers)# match direction ?
        egress   Match egress flows
        ingress  Match ingress flows
```

- The direction match statement added
- The “direction” is a new information element
  - Egress value added in the template
  - Egress value not added for the aggregation caches
  - Existing ingress templates are not modified

NetFlow Dynamic Top Talkers

New

- Somehow similar to the top talkers
  - But dynamic, done on the fly with show commands
  - But does not require modifications to the router config
  - But does not create a new cache
  - But no available with the MIB—obviously
- Even more useful than top talkers for security
- “show ip flow top” command:
  - show ip flow top <N> <aggregate-field> <sort-criteria> <match-criteria>
- Introduced in 12.4(4)T on the software based routers (7500 and below)

NetFlow Dynamic Top Talkers Examples

New

- Top ten protocols currently flowing through the router:

```
Router# show ip flow top 10 aggregate protocol
```

- Top ten IP addresses which are sending the most packets

```
Router# show ip flow top 10 aggregate source-address sorted-by packets
```

- Top five destination addresses to which we're routing most traffic from the 10.10.10.0/24 prefix

```
Router# show ip flow top 5 aggregate destination-address match source-prefix 10.10.10.0/24
```

- 50 VLAN's that we're sending the least bytes to:

```
Router# show ip flow top 50 aggregate destination-vlan sorted-by bytes ascending
```

- Top 20 sources of 1-packet flows:

```
router# show ip flow top 50 aggregate source-address match packets 1
```

Flexible Flow Record—Key Fields for Security

IPv4	
IP (Source or Destination)	Payload Size
Prefix (Source or Destination)	Packet Section (Header)
Mask (Source or Destination)	Packet Section (Payload)
Minimum-Mask (Source or Destination)	TTL
Protocol	Options
Fragmentation Flags	Version
Fragmentation Offset	Precedence
ID	DSCP
Header Length	TOS
Total Length	

Routing
Destination AS
Peer AS
Traffic Index
Forwarding Status
Is-Multicast
IGP Next Hop
BGP Next Hop

Flow
Sampler ID
Direction

Interface
Input
Output

Transport	
Destination Port	TCP Flag: ACK
Source Port	TCP Flag: CWR
ICMP Code	TCP Flag: ECE
ICMP Type	TCP Flag: FIN
IGMP Type	TCP Flag: PSH
TCP ACK Number	TCP Flag: RST
TCP Header Length	TCP Flag: SYN
TCP Sequence Number	TCP Flag: URG
TCP Window-Size	UDP Message Length
TCP Source Port	UDP Source Port
TCP Destination Port	UDP Destination Port
TCP Urgent Pointer	

Flexible Flow Record
Non-Key Fields for Security

Counters
Bytes
Bytes Long
Bytes Square Sum
Packet
Packet Long

Timestamp
sysUpTime First Packet
sysUpTime First Packet

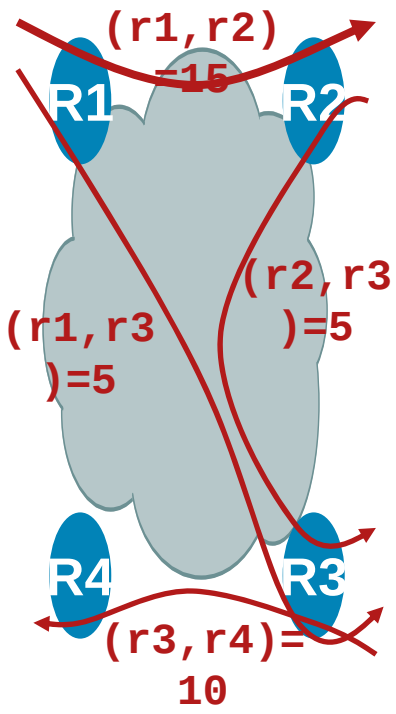
IPv4
Total Length Minimum
Total Length Maximum
TTL Minimum
TTL Maximum

- Plus any of the potential “key” field: will be the value of the first packet in the flow

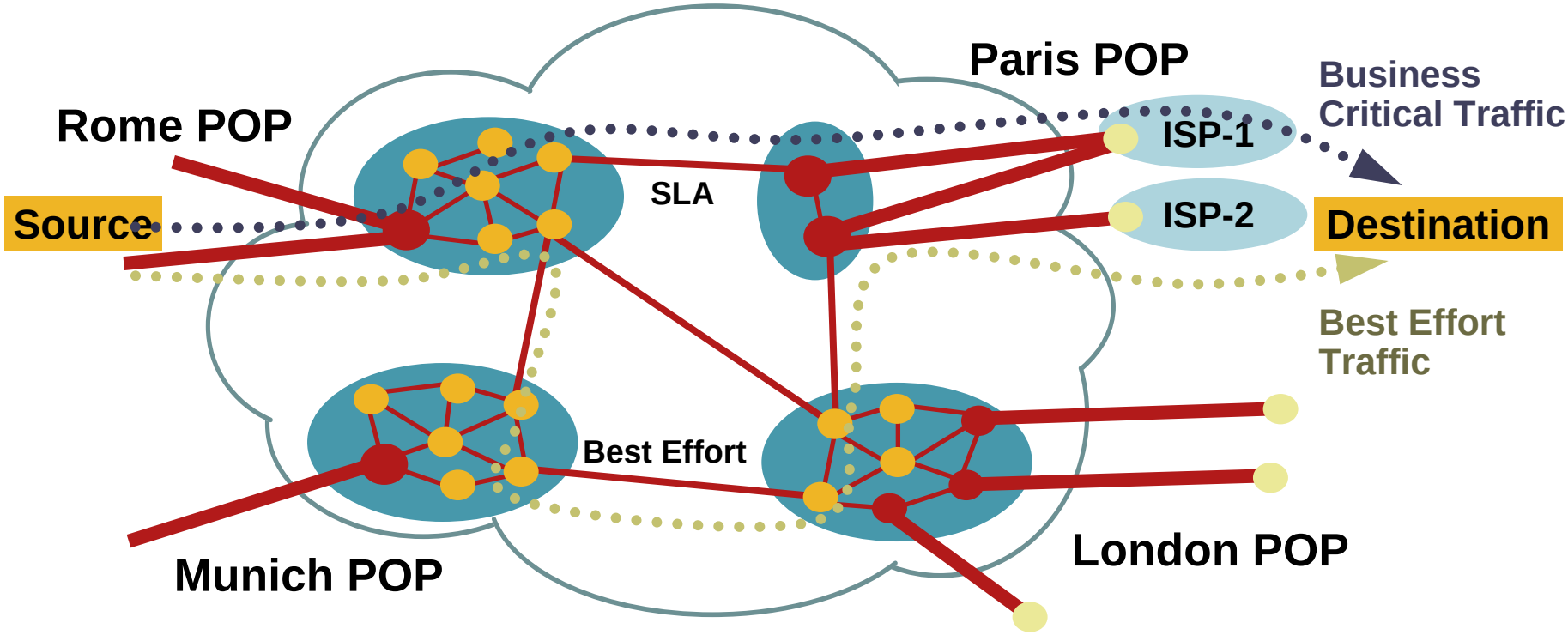
NetFlow for Capacity Planning

What Is the Traffic Matrix?

From/to	R1	R2	R3	R4
R1	0	15	5	0
R2	0	0	5	0
R3	0	0	0	10
R4	0	0	0	0



The Core Traffic Matrix
Traffic Engineering and Capacity Planning



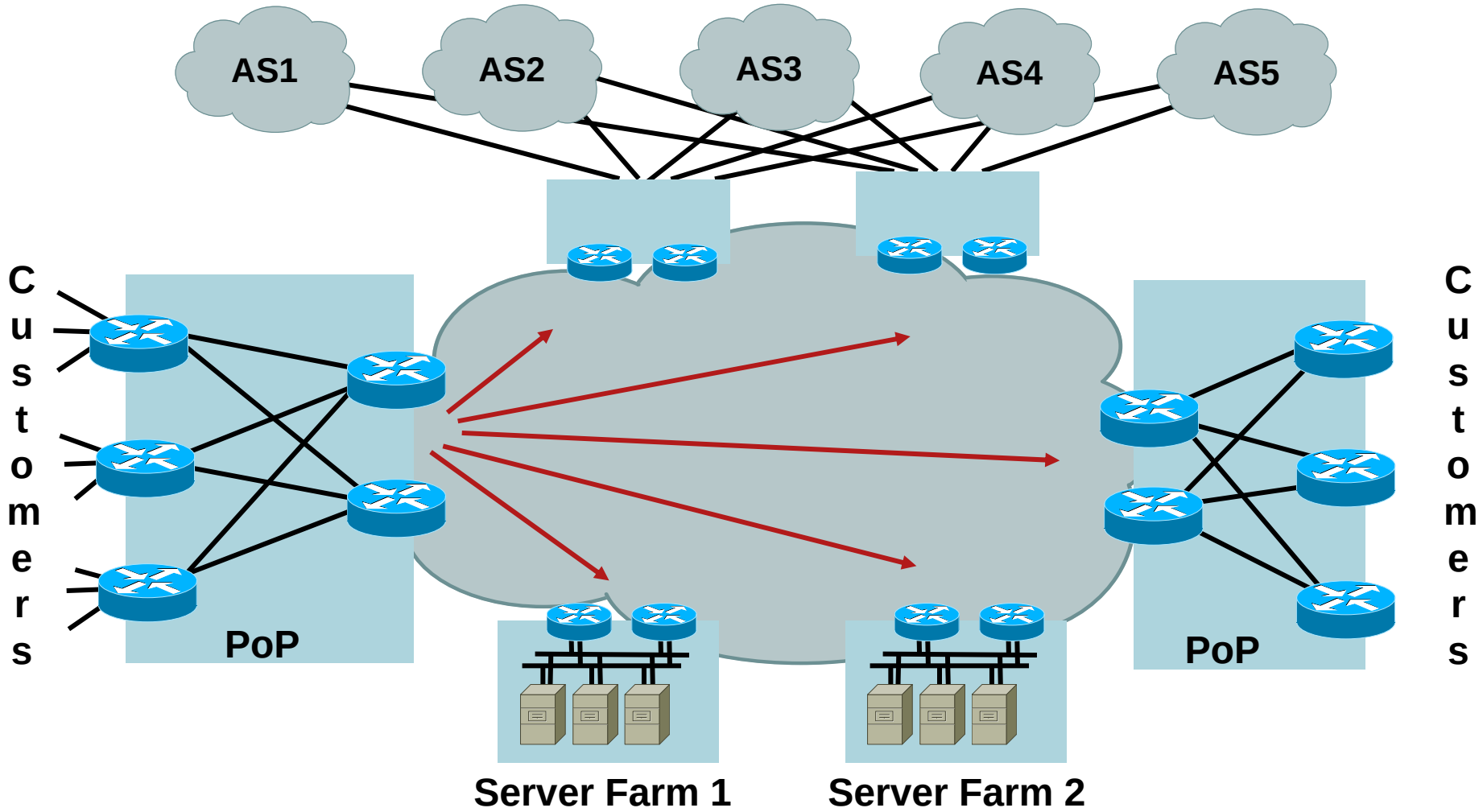
	Rome Exit Point	Paris Exit Point	London Exit Point	Munich Exit Point
Rome Entry Point	NA (*)	…Mb/s	…Mb/s	…Mb/s
Paris Entry Point	…Mb/s	NA (*)	…Mb/s	…Mb/s
London Exit Point	…Mb/s	…Mb/s	NA (*)	…Mb/s
Munich Exit Point	…Mb/s	…Mb/s	…Mb/s	NA (*)

(*) Potentially Local Exchange Traffic

Core Capacity Planning
The Big Picture

1. The ability to offer SLAs is dependent upon ensuring that core network bandwidth is adequately provisioned
2. Adequate provisioning (without gross over provisioning) is dependent upon accurate core capacity planning
3. Accurate core capacity planning is dependent upon understanding the core traffic matrix and flows and mapping these to the underlying topology
4. A tool for “what if” scenarios

We Need the Core Traffic Matrix

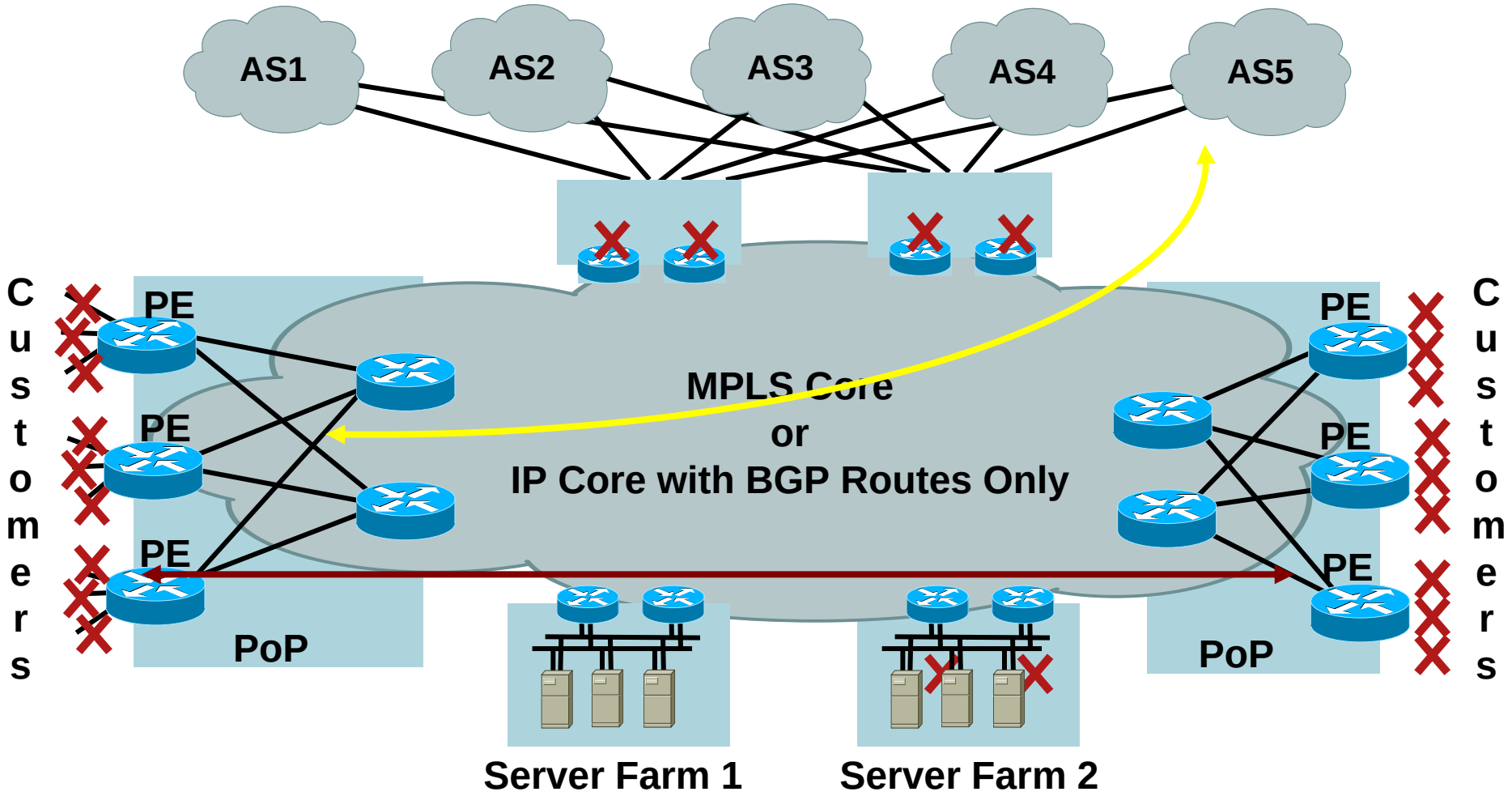


"PoP to PoP": Access Router or Core Router

NetFlow BGP Next Hop TOS Aggregation

- Lets you measure network traffic on a per BGP next hop basis, per TOS
- Lets you track which service provider the traffic is going through (exit point)
- Configure on ingress interface
- Leverages the new NetFlow version 9 export format
- Support with sampled and non-sampled NetFlow
- 12.0(26)S, 12.2(18)S and 12.3 on the software based routers (7500 and below)
- 12.0(27)S for the 12000

BGP Next Hop TOS Aggregation Typical Example



Internal Traffic: “PoP to PoP”

External Traffic Matrix PoP to BGP AS

NetFlow BGP Next Hop TOS Aggregation Flow Keys

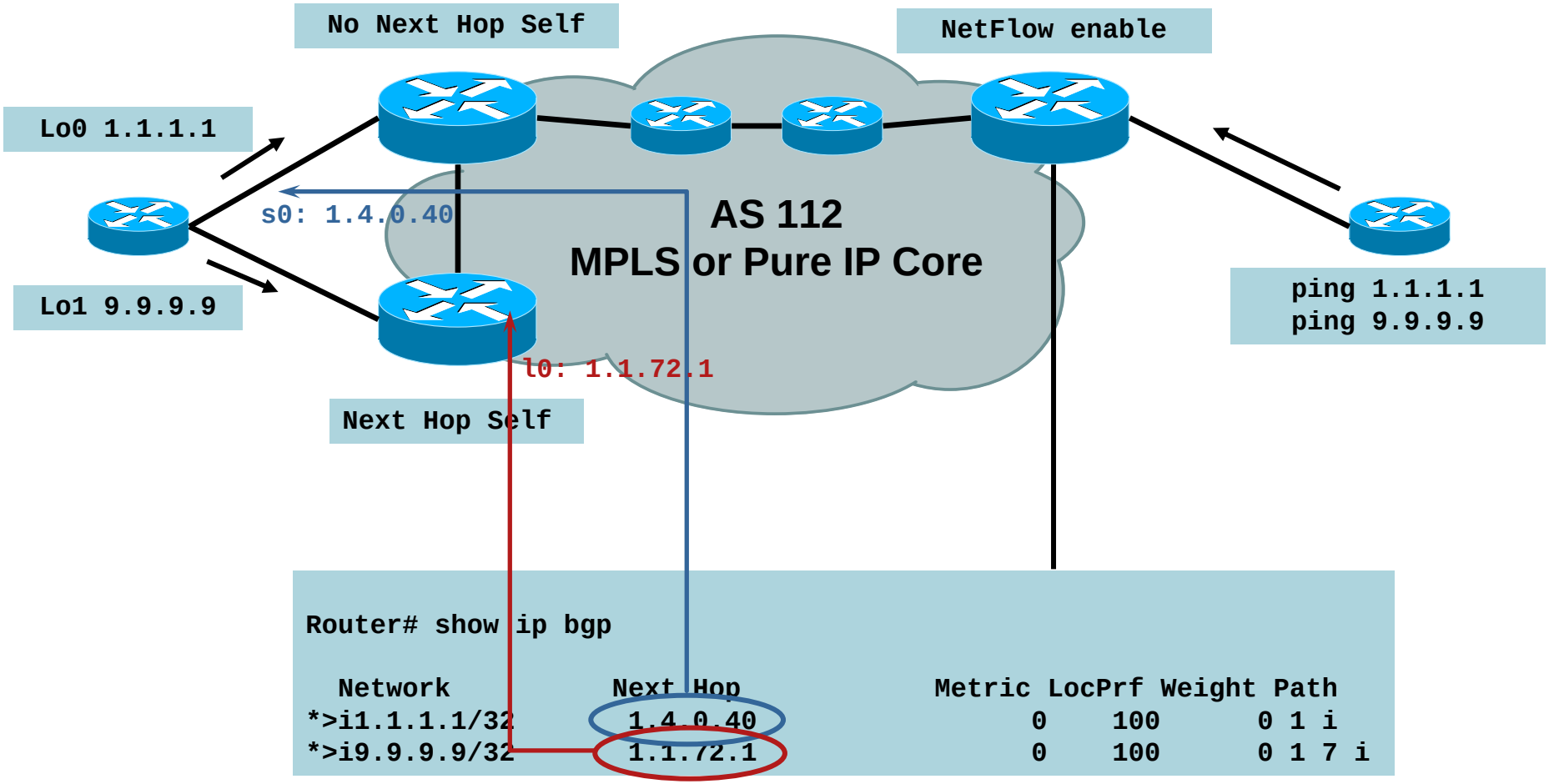
Key Fields (Uniquely Identifies the Flow)	Additional Export Fields
• Origin AS	• Flows
• Destination AS	• Packets
• Inbound Interface	• Bytes
• Output Interface	• First SysUptime
• ToS/DSCP (*)	• Last SysUptime
• Next BGP Hop	

(*) Before Any Recoloring

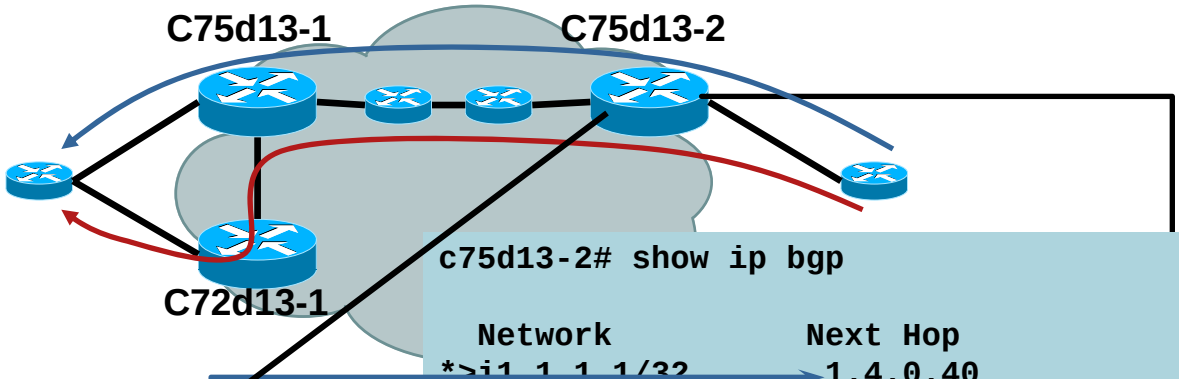
NetFlow BGP Next Hop TOS Aggregation Configuration

```
Router (config) # ip flow-export version 9 [origin-as | peer-as]
                  [bgp-nexthop]
Router (config) # ip flow-export destination <dest IP> <dest udp-port>
Router (config) # ip flow-export source <interface>

Router (config) # ip flow-aggregation cache bgp_nexthop_tos
Router (config-flow-cache)# export destination <dest IP > <dest udp-port>
Router (config-flow-cache)# enabled

Router (config-if)# ip flow ingress
```

NetFlow BGP Next Hop TOS Aggregation Testing



NetFlow BGP Next Hop TOS Aggregation Testing




```
c75d13-2# show ip bgp

  Network          Next Hop            Metric LocPrf Weight Path
*>i1.1.1.1/32      1.4.0.40                 0    100      0 1 i
*>i9.9.9.9/32      1.1.72.1                 0    100      0 1 7 i
```

For a Ping to 1.1.1.1

```
show ip cache verbose flow aggregation bgp-nexthop-tos

Src If         Src AS  Dst If         Dst AS  TOS Flows   Pkts  B/Pk  Active
BGP NextHop
Et1/0/1         2      Et1/0/2         1      00     1      5    100     0.0
BGP  1.4.0.40

Src If         Src AS  Dst If         Dst AS  TOS Flows   Pkts  B/Pk  Active
BGP NextHop
Et1/0/1         2      Et1/0/2         1      00     1      5    100     0.0
BGP  1.1.72.1
```

For a Ping to 9.9.9.9

Core Traffic Matrix with Flexible NetFlow

Key Fields (Uniquely Identifies the Flow)	Additional Export Fields
~~Origin AS~~	~~Flows~~
Destination AS	~~Packets~~
Inbound Interface	Bytes
~~Output Interface~~	First SysUptime
ToS/DSCP (*)	Last SysUptime
Next BGP Hop	

(*) Before Any Recoloring

- Less flow records, less CPU
- Potentially higher sampling rate for a better accuracy

Core Traffic Matrix with Flexible NetFlow Configuration Example

```
flow record traffic-matrix-record
   match routing destination as
   match interface input
   match ipv4 dscp
   match routing next-hop address ipv4 bgp
   collect counter bytes long
   collect timestamp sys-uptime first
   collect timestamp sys-uptime last

flow monitor traffic-matrix-monitor
   record traffic-matrix-record
   cache entries 10000
   cache type normal
   exporter capacity-planning-collector

interface pos3/0
   ip flow monitor traffic-matrix-monitor
```

Note: Export Less Flow Records with a Permanent Cache

Flexible NetFlow Cache
Improve BGP Policy Accounting Example

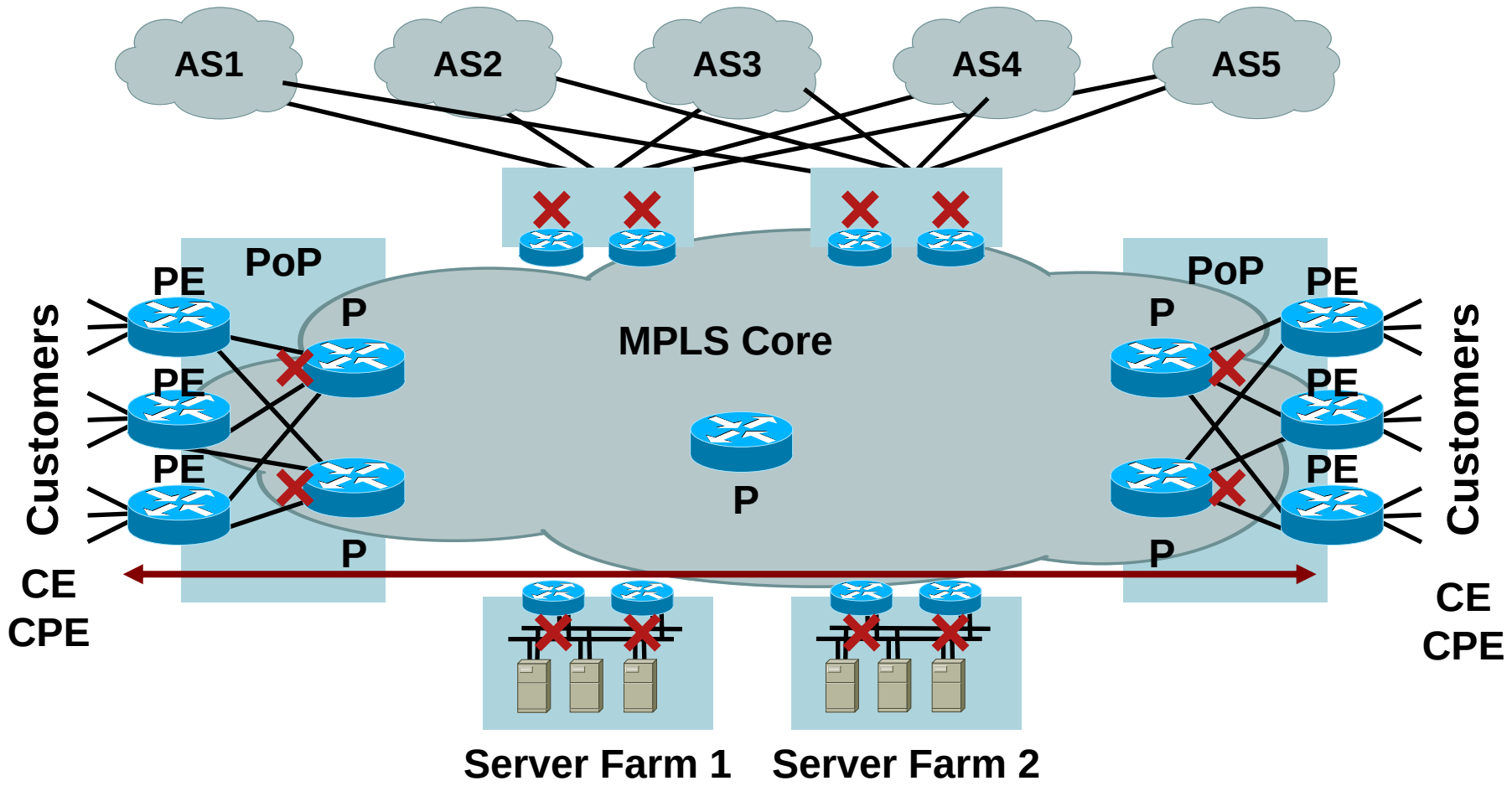
```
flow record traffic-matrix-record
   match routing destination traffic-index
   match interface input
   collect counter bytes long
   collect counter packets long

flow monitor traffic-matrix-monitor
   record traffic-matrix-record
   cache entries 1000
   cache type permanent
   cache timeout update 3600
   exporter capacity-planning-collector

interface pos3/0
   ip flow monitor traffic-matrix-monitor
```

Permanent Cache, with a Record Sent Every Hour

MPLS Aware NetFlow Description

- Provides flow statistics per MPLS and IP packets
  - MPLS packets:
    - Labels information
    - And NetFlow v5 fields for underlying IP packet
  - IP packets:
    - Regular IP NetFlow records
- Leverages the new NetFlow version 9 export format
- Configure on ingress interface
- Supported on sampled/non-sampled NetFlow
- 12.0(26)S1, 12.2(18)S and 12.3 on the software based routers (7500 and below)
  - 12000: 12.0(24)S, 12.2(18)S and 12.3

MPLS Aware NetFlow
The Core Traffic Matrix



Internal Traffic: “PoP to PoP”

External Traffic Matrix PoP to BGP AS: not available

MPLS Aware NetFlow Top Label Aggregation

Key Fields (Uniquely Identifies the Flow)

- Input interface (ifIndex)
- The top incoming MPLS labels with experimental bits and end-of-stack bit

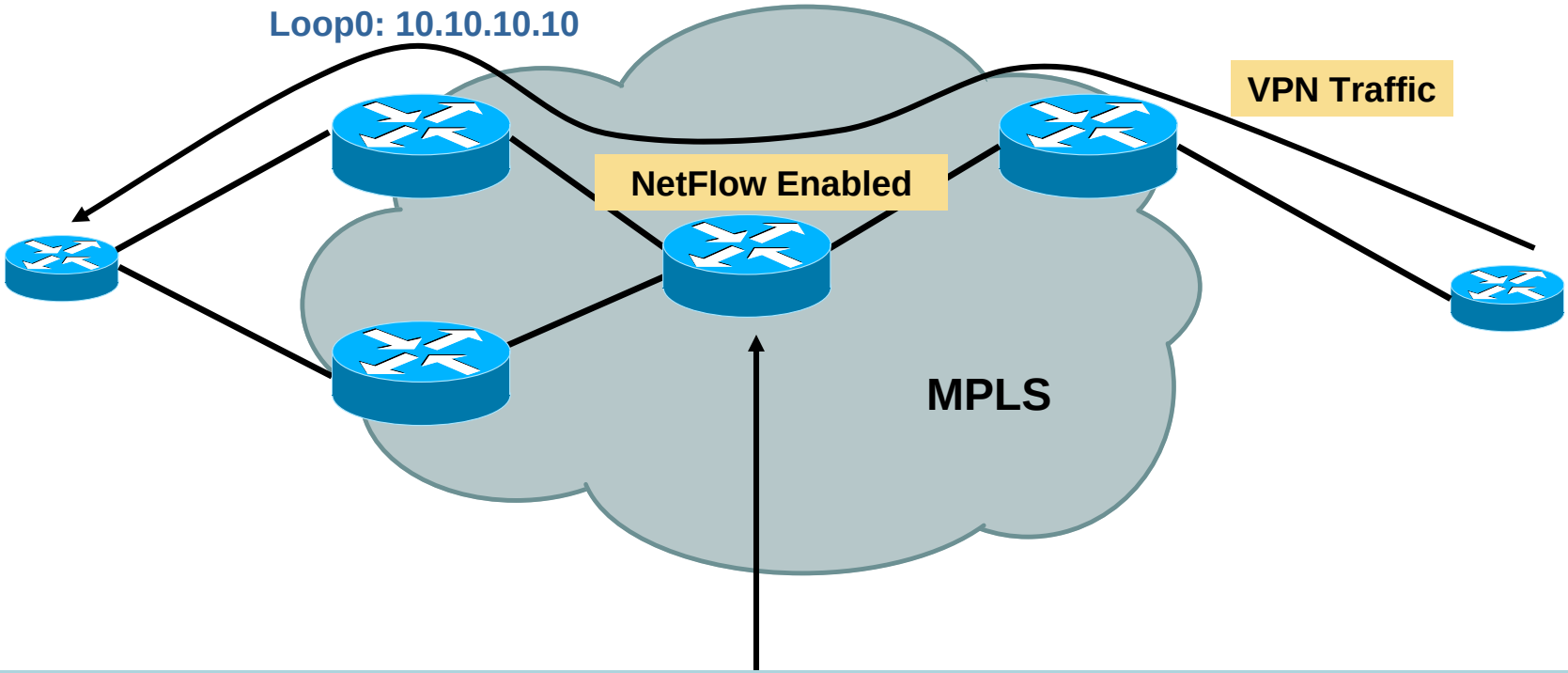
Additional Export Fields

- Flows
- Packets
- Bytes
- First timestamp (SysUptime)
- Last timestamp (SysUptime)
- Output interface
- NetFlow version five fields of the underlying IP packet (TCP flags, etc.)
- Type of the top label: LDP, BGP, VPN, ATOM, TE tunnel MID-PT, unknown
- The forwarding equivalent class mapping to the top label

MPLS Aware NetFlow
Top Label Aggregation Configuration

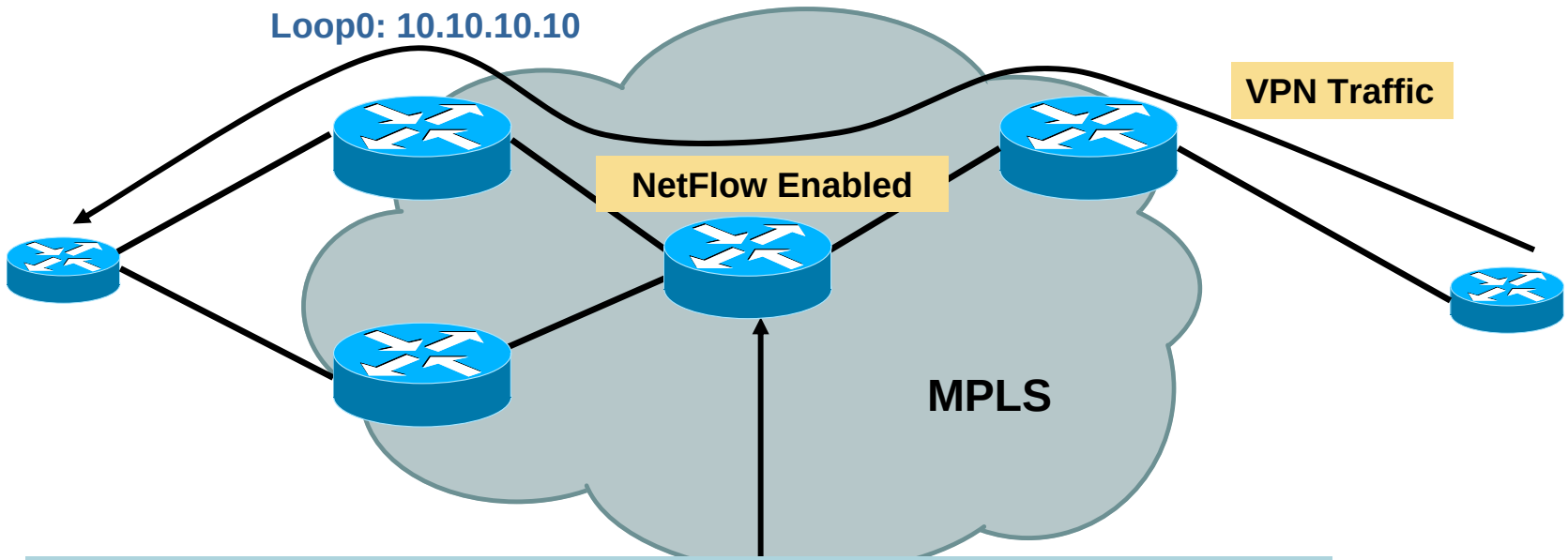
```
Router(config)#ip flow-cache mpls label-positions 1
               no-ip-fields mpls-length
Router(config-if)# ip route-cache flow sampled
Router(config)# ip flow-export version 9
Router(config)# ip flow-export template options export-stats
Router(config)# ip flow-export template options sampling
Router(config)# ip flow-sampling-mode packet-interval 100
```

NetFlow MPLS Aware Testing




```
Router# show mpls forwarding 10.10.10.10
Local   Outgoing   Prefix              Bytes tag      Outgoing    Next Hop
tag     tag or VC  or Tunnel Id        switched       interface
486     Pop tag    10.10.10.10/32  1696244602516  PO3/0       point2point
```

NetFlow MPLS Aware Testing




```
Router# show ip flow verbose cache
…
SrcIf   SrcIPaddress DstIf   DstIPaddress   Pr   TOS   Flgs   Pkts
Port   Msk   AS   Port   Msk   AS   NextHop   B/Pk   Active

PO2/0   0.0.0.0       PO3/0   0.0.0.0        00   00    10     1729
0000   /0    0    0000   /0    0    0.0.0.0    792   14.6
Pos:Lbl-Exp-S 1:486-4-0 (LDP/10.10.10.10)
```

Exported as 7784, EXP in NFC 5.0

Platforms Specific

NetFlow Implementation

- The software based routers
  - 7500, 7200, 3800, 2800, 1800, 800
- Metering process in hardware
  - Catalyst 6500/ 7600 Router
  - 12000 engine 3 and 5
  - 10000

Configuring NetFlow
Cisco IOS on 7600/Cisco Catalyst 6500

```
C6500(config)#mls netflow
```

Enable NetFlow

Set the Flow Mask

```
C6500(config)#mls flow ip ?
  destination                     destination flow keyword
  destination-source              destination-source flow keyword
  full                            full flow keyword
  interface-destination-source    interface-destination-source flow
keyword
  interface-full                  interface full flow keyword
  source                          source only flow keyword

C6500(config)#mls nde sender version ?
  5
  7
```

Set the NetFlow Record Version on PFC

Populate Interface Field in NDE Packet

```
C6500(config)#mls nde interface
```

Change Default HW Timer

```
C6500(config)#mls aging normal 32
```

Destination for PFC/MSFC Exports

```
C6500(config)#ip flow-export destination 10.66.231.10

C6500(config)#interface g1/1
C6500(config-if)#ip route-cache flow
```

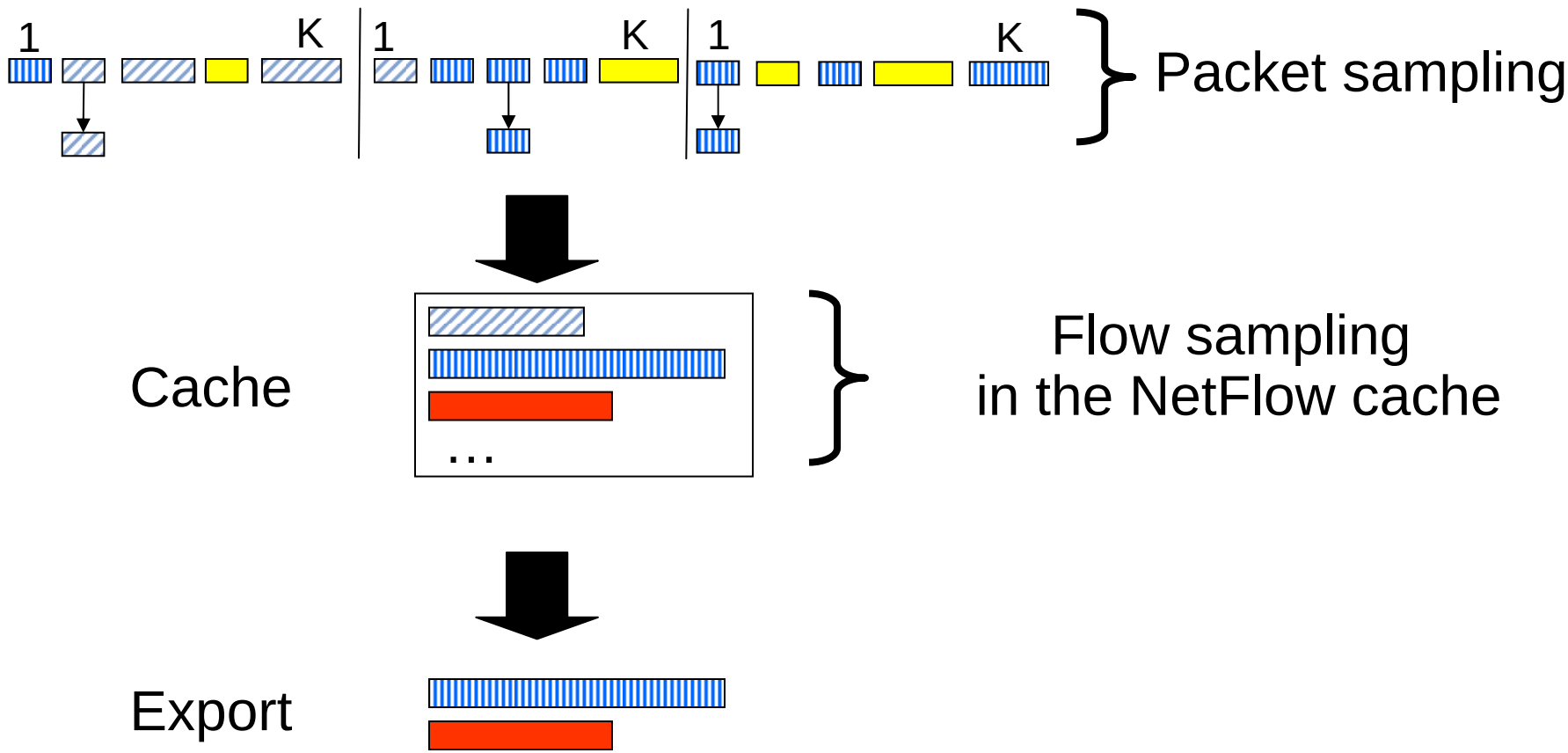
Software Flows Interface Capture

Cisco Catalyst 6500 and Cisco 7600 Capacities Across the Supervisor Family

- Each of the supervisors support for NetFlow yields a different number of flows that can be stored in the NetFlow tables

	Table Size	Hash Efficiency	Effective Size	Hash Key Size
Sup2	128K	25%	32K	17 Bits
Sup720	128K	50%	64K	36 Bits
Sup720-3B	128K	90%	115K	36 Bits
Sup720-3BXL	256K	90%	230K	36 Bits
Sup32-8GE	128K	90%	115K	36 Bits
Sup32-10GE	128K	90%	115K	36 Bits
Sup720-10GE-3C	128K	90%	115K	36 Bits
Sup720-10GE-3CXL	256K	90%	230K	36 Bits

Packet Sampling Versus Flow Sampling



Sampling NetFlow on the Cisco Catalyst Flow Sampling

- Not packet sampling but flow sampling

  Reason: NetFlow in hardware on the Cisco Catalyst

- Time based flow sampling

  Take a snapshot of the NetFlow cache at different time

- Packet based flow sampling

  At each Delta_T, export the flows with minimum values of packet M

  For flows with packet count < M, packet counts will be summed up, and one “flow” will be sent

Cisco Catalyst 6500 and Cisco 7600 Series Versions and Features

- PFC2 source/destination interface information: 12.1(13)E1, hybrid 6.3(6)
- PFC2 source/destination AS information: 12.1(13)E1
- PFC2 support for v5 NetFlow data export: 12.1(13)E1, hybrid 7.5(1)
- Version 8 in native mode, 12.2(14)SX
- Dual export support sup2: 12.2(17d)SXB
- Input ToS field: PFC3b and 3bXL (sup720) cards
- L2 switched traffic (vlan x to vlan x) support: hybrid 7.2(1)
- Per vlan NetFlow: hybrid OS 8.4
- NetFlow multicast with NetFlow version 9: 12.2(18)SXF
- NetFlow BGP next hop with NetFlow version 9: 12.2(18)SXF
- In development, NetFlow and IPv6 (first half 2007)

Cisco 10000

- 12.2.31-SB2
  - NetFlow version 9
  - Egress Netflow with EXP capture (MPLS egress NetFlow)
  - Egress Sampled Netflow (random sampling)
  - BGP next hop

NetFlow Ongoing Developments

Sampled NetFlow

- Capacity planning may not need every packet per flow
- Sampling will reduce CPU consumption
- Random (select packet to export per statistical principles)

  Cisco IOS software releases 12.0(26)S, 12.2(18)S, and 12.3(1)T

  Software platforms 7xxx, 37xx, 36xx, 26xx

  Cisco 12000 series: deterministic sampling today

  Cisco Catalyst 6000: no packet sampling but flow sampling
- Deterministic/Random with flexible NetFlow

Accuracy of (Packet) Sampled NetFlow Research Project

- What is the accuracy of sampled NetFlow? Is the estimated number of bytes per flow record accurate? Which sample rate should be used?
- Developed an mathematical model

  This model is only valid for random sampled NetFlow

  Systematic sampled NetFlow would require some knowledge about the traffic patterns

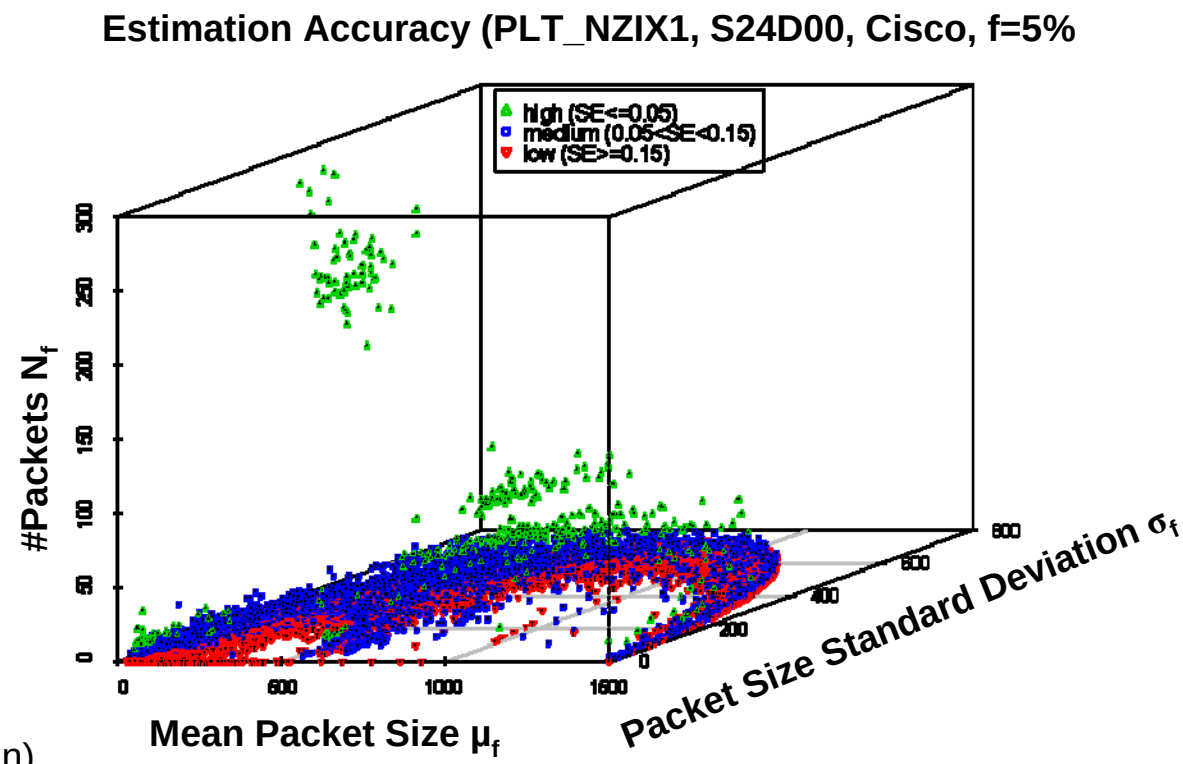
$$\mathrm{StdErr}_{rel}[\hat{\mathrm{Sum}}_f] = \frac{\mathrm{StdErr}_{abs}[\hat{\mathrm{Sum}}_f]}{\mathrm{Sum}_f} = \frac{\sqrt{\frac{N^2}{n} \cdot \left(\sigma_{x_f}^2 \cdot P_f + \mu_{x_f}^2 \cdot (P_f - P_f^2)\right)}}{N_f \cdot \mu_{x_f}}$$

- Square sum of bytes available in Flexible NetFlow

  “collect counter bytes squared long” in the CLI

Accuracy of (Packet) Sampled NetFlow Research Project

- Empirical testing with real live testing
  - Mathematical model validity
  - Mathematical model assumptions
  - Results confidence interval
  - Graph the results
- Higher accuracy for flows with
  - Many packets
    - Flow proportion is high
    - Observe longer (and characteristics remain)
  - Large packet size mean
  - Small packet size variation
- Paper will be published soon



IETF: IP Flow Information Export WG (IPFIX)

- RFC3954 “Cisco Systems NetFlow Services Export Version 9”

  NetFlow patent: intellectual property right statement on the IETF website

- IPFIX is an effort to:

  Define the notion of a “standard IP flow”, along with data encoding for IP flows

  http://www.ietf.org/html.charters/ipfix-charter.html

- RFC3917 “Requirements for IP Flow Information Export”

  Gathers all IPFIX requirements for the IPFIX evaluation process

- RFC3955 “Evaluation of Candidate Protocols for IPFIX”

IETF: IP Flow Information Export WG (IPFIX)

- IPFIX protocol specifications
  - Changed in terminology but same principles as NetFlow version 9
  - Improvements versus NetFlow version 9: SCTP-PR, security, variable length information element, IANA registration, etc.
  - Generic streaming protocol, not flow-centric anymore
  - Security:
    - Threat: confidentiality, integrity, authorization
    - Solution: DTLS on PR-SCTP
- IPFIX information model
  - Most NetFlow version 9 information elements ID are kept
  - Proprietary information element specification

IETF: IPFIX Status

- All IPFIX drafts transmitted to the IESG (Internet engineering task force)
  - IPFIX Protocol draft in the RFC-Editor queue
  - IPFIX Architecture draft: one more correction and then RFC-editor queue
  - IPFIX Information: some comments from the IESG
- IPFIX Prototype done during interop
- Foreseen in IOS in first half 2008
  - Is it important to you?

IETF: Packet Sampling WG (PSAMP)

- PSAMP is an effort to:

  Specify a set of selection operations by which packets are sampled, and describe protocols by which information on sampled packets is reported to applications

- Sampling and filtering techniques for IP packet selection

  To be compliant with PSAMP, we must implement at least one of the mechanisms: sampled NetFlow, NetFlow input filters are already implemented

- PSAMP protocol specifications

  Agreed to use IPFIX for export protocol

- Information model for packet sampling export

  Extension of the IPFIX information model

Conclusion

NetFlow Summary and Conclusion

- NetFlow is a mature Cisco IOS feature (in Cisco IOS since 1996)
- NetFlow provides input for accounting, performance, security, and billing applications
- NetFlow has IETF and industry leadership
- NetFlow v9 eases the exporting of additional fields
- A lot of new features have been added
- Flexible NetFlow is a major enhancement
- Stay tuned for more ☺

References

- NetFlow

  http://www.cisco.com/go/netflow

- Cisco network accounting services

  Comparison of Cisco NetFlow versus other available accounting technologies

  http://www.cisco.com/warp/public/cc/pd/iosw/prodlit/nwact_wp.htm

- Cisco IT case study

  http://business.cisco.com/prod/tree.taf%3Fasset_id=106882&IT=104252&public_view=true&kbns=1.html

- A complete white paper

  http://www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/netflsol/nfwhite.htm

- NetFlow product manager, Jean Charles Griviaud, jgriviau@cisco.com

Meet the Experts
Management & Operations

- Benoit Claise
  Distinguished Service Engineer
- Bruno Klauser
  Consulting Systems Engineer
- Emmanuel Tychon
  Technical Marketing Engineer
- Ralph Droms
  Technical Leader
- Stephen Mullaney
  Technical Marketing Engineer
- Stuart Parham
  Consulting Systems Engineer

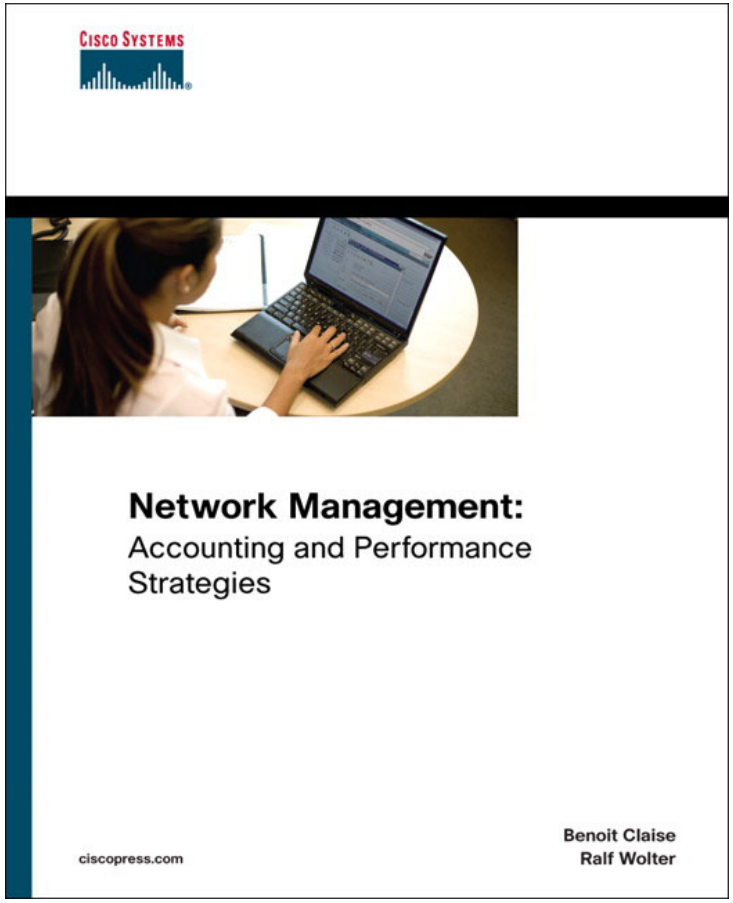





Recommended Reading

BRKNMS - 3006

- Network Management: Accounting and Performance Strategies (Jul 07)



Available in the Cisco Company Store

Q and A

Appendix

Embedded Agents
Accounting – NetFlow evolution in IOS-XR

- R 3.2: support of IPv4 NetFlow on CRS-1.
- R 3.3: support of IPv4 NetFlow on c12k and support of sub-interfaces and Bundles on CRS-1.
- R 3.3.1: support of MPLS-aware NetFlow on CRS-1 only.
- **R 3.4**: XML support for config and some show and clear cmds. Support for up to 8 exporters per flow monitor-map.
- R 3.4.1: MPLS NetFlow extended
- **What is not supported**:
  - **Full (non-sampled) mode**
  - **IPv6 traffic**
  - **Deterministic sampling algorithm**
  - **V5, v8 NetFlow export formats**
  - **Aggregation schemes**